Kandidatexjobb i elektroteknik 2021

Kungliga Tekniska högskolan, Stockholm

PREFACE

This book contains all project reports from the EF112X bachelor thesis course in spring 2021. The course is directed towards students from the engineering programme within the EECS school at KTH, but is also open to physics, vehicle engineering and energy and environment students from KTH. This spring 68 electrical engineering students, 37 physics, 18 vehicle engineering, 5 energy & environment and 2 computer science students participated with 67 different projects, most of which have been done in two-person groups. Due to the Corona situation in spring 2021, the entire course had to be done online, including digital lectures and a digital project presentation day. Both, students and supervisors did an extraordinary job in organizing themselves, keeping online contact and sticking to the project plans, such that the work could be finished in time resulting in the scientific project reports presented in this book.

The bachelor thesis course spans each year from mid January to mid May and gives in total 15 credits. It consists of the individual project work of each student group, and a series of seminars, workshops and computer labs. The course ends with a project presentation day that is organized in a similar way as a scientific conference, including moderators and student opponents. In the seminars and workshops, the students train how to organize their work, how to find, judge and cite other's work, and how to present their project results orally as well as in written reports. In addition, the students train to reflect critically on their role as engineers, and how their project context may affect society and environment.

The reports presented in this book cover a wide range of topics that are grouped into 16 larger contexts. Each context in this book contains an introduction and reports. The introduction texts consist of a popular description, a summary of the project results and a reflection about the context's importance for a sustainable society. Each context introduction has been written by those students that work on projects within that context. This year's contexts cover the following topics:

CONTEXT A: Automatic Car Following and Platooning
CONTEXT B: Autonomous Robotic Systems
CONTEXT C: Learning in Dynamical Systems
CONTEXT D: The CO2-free Power System
CONTEXT E: HVDC Grids
CONTEXT F: Power System Control
CONTEXT G: Fusion – the Sun's Energy Source on Earth
CONTEXT H: Observations in Space
CONTEXT I: Suborbital Free Flyer for Near-Earth Space Research
CONTEXT J: Design and Testing of Novel Microwave & Antenna Technologies
CONTEXT K: Electrotechnical Multiphysics Simulation
CONTEXT L: AIoT - Artificial Intelligence and the Internet of Things
CONTEXT M: Information Engineering - Big Data and AI
CONTEXT O: Computational Brain Modelling & Brain-like Computing
CONTEXT P: Artificial Intelligence
CONTEXT R: Embedded Systems

In this course, the students chose among projects that have been proposed by supervisors in advance and are described in a so-called project catalogue. Each of the proposed projects in the catalogue has an identification number, which re-appears in the table of content of this book as well as in the header of the corresponding report. The most popular projects have been done by several groups, which is indicated by similar identification numbers (e.g., C2A and C2B). Other numbers do not appear at all, as those projects have not been selected by any students this year. The project titles in the table of content appear in English or Swedish, depending on the language in which the corresponding report has been written.

This book would not have been possible without the hard work of all participating students their supervisors and supervisor assistants. Many thanks to the teachers in this course: Joakim Lilliesköld (work plan), Martin Lindberg and assistants (latex lab), and Anna Herland (source critics and review). I myself took care of the project selection, course intro, report and popular writing, ethics and sustainability seminars, the organization of the workshop days and the final presentation day. My gratitude goes especially to Kristin Linngård for her invaluable administrative help!

Anita Kullen
Course responsible for the EF112X bachelor thesis course
Stockholm, September 10, 2021

TABLE OF CONTENTS

CONTEXT A: Automatic Car Following and Platooning 7

A1a. Model Predictive Control for Vision-Based Platooning Utilizing Road Topography 11
A1b. Stability of a Vision Based Platooning System 19
A2a. Splitting and Merging of Platoons With the Help of PID Control 27
A2b. Splitting a Platoon Using Model Predictive Control 35

CONTEXT B: Autonomous Robotic Systems 45

B1. Motion Planning for Aggressive Flights of an Unmanned Aerial Vehicle 49
B2. Cooperative Control of Autonomous Ground Vehicles 59
B3. Multi-Robot Motion Planning With Control Barrier Functions for Signal Temporal Logic Tasks 67

CONTEXT C - Part I: Learning in Dynamical Systems 77

C1a. Stock Price Prediction Using SVR with Stock Price, Macroeconomic and Microeconomic Data 85
C1b. Using A Hidden Markov Model as a Financial Advisor 93
C2a. Asynchronous Advantage Actor-Critic and Flappy Bird 101
C2b. Control of an Inverted Pendulum Using Reinforcement Learning Methods 111
C3a. Warehouse Optimization by Multi-Agent Rollout Algorithms 119
C3b. Distributed Deep Reinforcement Learning for a Multi-Robot Warehouse System 125

CONTEXT C - Part II: Learning in Dynamical Systems XXX

C4a. Machine Learning Methods for Predicting Trading Behaviour of an Actively Managed Mutual Fund 133
C4b. Inversion of Markowitz Portfolio Optimization to Evaluate Risk 141
C5a. Generation and Detection of Adversarial Attacks for Reinforcement Learning Policies 147
C5b. Robust/Adversarial Deep Reinforcement Learning 155

CONTEXT D: The CO2-free Power System 167

D1. Voltage Deviations in a Power System 171
D2. Implementation of a Capacity Market in Sweden 189
D3. Modellering av vattenkraft i Spine: En studie om gränserna för vattenkraften som reglerande energikälla 199

CONTEXT E: HVDC Grids 207

E2. Protection of HVDC Grids Against Blackouts (Simulation) 211

CONTEXT F: Power System Control 219

F1. Frequency Stability in Future Low Inertia Power Systems With Battery Support 223
F2. Tuning a Power System Stabilizer to Damp Out Power Oscillations 231
F3. Design of a Future Residential Hybrid Microgrid 237

CONTEXT G: Fusion – the Sun's Energy Source on Earth 245

G1. Data Processing in Accelerator-Based Analysis of Wall Materials From Controlled Fusion Devices 249
G3. Modeling of RF Heating in the JET Tokamak 257

CONTEXT H: Observations in Space 269

H2. Electron Acceleration at Earth Bow Shock 271
H3. Search for Water Plumes on Jupiter's Moon Europa 277

CONTEXT I: Suborbital Free Flyer for Near-Earth Space Research 285

I1. High Altitude Glider Solution for Returning From Space 289
I2. Deployable wing structure for the FFU 305
I3. Simulation and Control System Design for Autonomous Gliding to a Given Location 317
I4. Power and Electronics in Autonomous Glider for Sounding Rocket Experiments 335

CONTEXT J - Part I: Novel Microwave & Antenna Technologies 345
J1a. 3D-Printed Geodesic Luneburg Lens Antenna With Novel Patch Antenna Feeding 353
J1b. 3D-Printed Geodesic Reflective Luneburg Lens Antenna for X-Band 365
J2. Glide-symmetric Holey EBG Filter Using Multiple Unit cell Designs 375

CONTEXT J - Part II: Novel Microwave & Antenna Technologies XXX
J3a. Fully Metallic One Dimensional Uniform Tapered-Pin Leaky-Wave Antenna at 30 GHz 385
J3b. Millimeter-Wave Pencil Beam Leaky-Wave Antenna 395
J4a. Measurement of the Complex Permittivity of Phantoms for 5G/6G Compliance Tests 403
J4b. Simulation and Measurement of Body Absorption for 5G/6G Frequency Bands 411

CONTEXT K: Electrotechnical Multiphysics Simulation 417
K1. Simulations of Plasma Creating Electric Wind 419

CONTEXT L - Part I: AIoT - Artificial Intelligence and the Internet of Things 427
L2a. Activity Recogniton Using Accelerometer and Gyroscope Data From Pocket-Worn Smartphones 435
L2b. Using Machine Learning for Activity Recognition in Running Exercise 441
L3a. Water Anomaly Detection Using Federated Machine Learning 447
L3b. Federated Machine Learning for Water Monitoring Systems 457

CONTEXT L - Part II: AIoT - Artificial Intelligence and the Internet of Things 427
L6a. Hacking and Evaluating the Cybersecurity of an Internet Connected 3D Printer 465
L6b. IoT Security Assessment of a Home Security Camera 479
L7. Achieving Full Attack Coverage and Compiling Guidelines for enterpriseLang 489
L9. Evading Swipe Pattern Classifiers Using a Generative Adversarial Network 497

CONTEXT M - Part I: Information Engineering - Big Data and AI 507
M1. Variable Selection in High-Dimensional Data 515
M2. Compression and Distribution of a Neural Network With IoT Applications 527
M3. Modelling of the DNA Helix's Duration for Genome Sequencing 535

CONTEXT M - Part II: Information Engineering - Big Data and AI 507
M5. A Small Classification Experiment Between Dolls and Humans With CNN 543
M6. Evaluating Methods for Show-through or Bleed-through Cancellation 553
M7. Machine Learning-based Biometric Identification 569

CONTEXT O: Computational Brain Modelling & Brain-like Computing 577
O1. The Impact of Selective Plasticity Modulation on Simulated Long Term Memory 581
O2. Effects of Network Size in a Recurrent Bayesian Confidence Propagating Neural Network 593

CONTEXT P - Part I: Artificial Intelligence 605
P1a. Knowledge Based Strategies in Grid-Based Pursuit-Evasion Games of Imperfect Information 613
P1b. Multi-Agent Games of Imperfect Information: Algorithms for Strategy Synthesis 627
P2a. En spelteoretisk AI för Stratego 639

CONTEXT P - Part II: Artificial Intelligence XXX
P2b. Game Theoretical AI Plays Strategy Board Game 647
P3a. Boosting CNN Performance in Digital Pathology Using Colour Normalisation and Ensembling 655
P3b. Assessing the Impact of Stain Normalization on a Cell Classification Model in Digital Histopathology 671

CONTEXT R: Embedded Systems 687
R1. Using Correlation Analysis to Locate Encryption Activity in Electromagnetic Side-Channels 691
R2. Design and Development of a Communication Middleware for Distributed Embedded Systems 699
R4. Telemetry System for Real-Time Monitoring of a Formula Student Electric Vehicle 709

CONTEXT A
AUTOMATIC CAR FOLLOWING AND PLATOONING

POPULAR DESCRIPTION

Drive closer to go further

Automated vehicles driving as close as 50 cm apart might seem scary, but it is possible! It is a technology that has the potential to revolutionize world wide transport. By working together, vehicles can safely travel with small distances to each other. This decreases the fuel consumption of trucks and improves traffic flow. A setup like this is called a platoon.

In recent years, the computing power and sensor information available for vehicles has grown rapidly. This allows smart communication between vehicles and opens up possibilities for smart route planning. By looking at multiple vehicles driving together as a single unit instead of individual vehicles, maneuvering and planning can be done together for the benefit of all vehicles involved.

If you've ever watched Tour de France, you might have noticed how the cyclists ride in a straight line. The leading cyclist, acting as a shield, reduces the air drag for the following cyclists and makes the ride easier for them. The same concept applies to vehicles in a platoon. When the trucks are driving closely behind each other, the first vehicle in the chain reduces the air resistance for the followers. Just like the cyclists can save energy, so can the trucks!

What will the future look like with platoons? We can imagine that we will have trains on our roads. Trucks that follow trucks like wagons follow wagons on a train. By making platooning more common on our roads we will also be able to leave the coming generations a greener future. Also, the companies will save money, we will have less traffic jams and spend less time on the roads. Do you feel urged to hop on the train?

SUMMARY OF PROJECT RESULTS

The concept of platooning refers to several vehicles driving closely together with a set of intervehicular spacings. The trailing vehicles are usually automated to some extent, while the leading vehicle can be manually or autonomously driven. The factors that are driving the development of the concept forward are many and include environmental, humanitarian and financial aspects.

Several solutions for platooning have been discussed and implemented in earlier research. The methods include different controllers, different ways of determining the positions of vehicles nearby and several other aspects. The groups in project A1 have focused on vision-based platooning while the projects in A2 have focused on splitting and merging vehicles in a platoon.

The most common platooning solutions use wireless communication between vehicles. When data is broadcasted wirelessly this comes with the risk of data interference and security concerns. A more secure solution could be the use of vision for determining each vehicle's position. The groups in project A1 have implemented vision-based platooning with the use of a camera and ArUco markers to track and adjust intervehicular spacings. The systems were first developed and tested with a simulation software and then on Small Vehicles for Autonomy (SVEA).

The groups A1a and A1b focused on two different approaches to broaden the research. How a vision-based platooning system can be used to minimize unnecessary accelerations and how the stability of a vision-based system compares to a cooperative communication system.

Project group A1a studied how vision-based platoons can benefit from utilizing road topography for enhanced performance. By accurately detecting the pose of an ArUco marker, the changes of the road topography could be detected. With this information, the platoon could perform real time adaptations, demonstrating the convenience of not having to rely on previously collected road data. A model predictive controller (MPC) was designed to minimize the motor and brake forces

while maintaining a platooning distance. The MPC was implemented on the SVEA and relies solely on data from ArUco markers to take advantage of the robustness that vision-based platooning offers.

As mentioned above, using vision-based platooning instead of communication-based has its benefits. However, communication enables a more stable system. A combination of the two systems could therefore lead to a more robust platooning solution. Based on this, project group A1b widened the scope of their study by investigating the stability of a vision-based platooning system compared to a cooperative communication-based system. To achieve vision-based platooning, an optimal-velocity-relative-velocity (OVRV) model was used. This enabled a smooth integration with the communication platooning that uses a cooperative OVRV model.

The project groups in project A2 have implemented controllers for splitting and merging vehicles in a platoon in a simulation environment. To solve this problem, the groups chose different controllers, namely proportional-integral-derivative (PID) controller and model predictive controller (MPC). A controller is an algorithm that compares a reference signal with the current output of a system and computes an input signal such that the output approaches the reference value. Both controllers solved the problem and managed the split under ideal conditions. Project group A2a, that has used a PID controller, introduced noise on the speed of the lead vehicle. Project group A2b introduced air drag to the system. Both controllers still managed the split but when the PID controller was used the system got more oscillatory while the MPC solved the air drag in a straightforward manner because of its predictive nature.

It is easier to add constraints to the system when using an MPC. Furthermore, the predictive nature of the MPC makes it more resistant to disturbances and thus making the MPC more reliable. Some constraints could still be added when using a PID controller. However, the PID controller is cheaper and significantly less computationally heavy compared to the MPC.
Both PID and MPC could manage a split under ideal conditions and with some disturbances added. There have not been any practical experiments on the created systems which is an area for future studies. In both A2 projects all the vehicles in the platoon were assumed to be autonomous. When a non-autonomous vehicle is near a platoon or tries to break the platoon it can cause problems since the autonomous vehicles motions are unpredictive. The results show that MPC and PID can be used in the future when most of the vehicles are autonomous but are not so safe to use when non-autonomous vehicles are combined in the traffic.

To conclude, automated car following and platooning has development potential and more research will have to be conducted. In project A1, both the performance of the MPC and a fallback system can be further developed. Regarding the fallback system, it would be beneficial to research and build on the integration between a camera based system and a communication based system. It could also be of interest to investigate further potential of the usage of a camera in terms of vision recognition for platooning. For the vision-based MPC, further work could examine to what degree this would benefit platoons of heavy-duty vehicles, and look more specifically at fuel saving implications. There is also the possibility of broadcasting road grade information to vehicles further back in a longer platoon. This could extend the prediction horizon while retaining the advantage of not relying on previously collected data.

For future studies in project A2 a system where there are more disturbances could be investigated. Furthermore, the system's response can be studied for the case that a vehicle merges into the platoon. For this, a smart algorithm beyond the control mechanism needs to be studied. Another aspect to look at in more detail is, how the system reacts when one or more of the vehicles in the platoon have a driver, i.e the system is not completely autonomous.

IMPACT ON SOCIETY AND ENVIRONMENT

Autonomous driving and platooning has many advantages and disadvantages regarding its impact on both the environment and society. One of the main advantages is the decreased fuel consumption. This is due to reduced air resistance associated with small intervehicular spacings. Autonomous vehicles in general also have this advantage, since the driving can be optimized with more data about other vehicles and road topography.

A platoon of vehicles drives closer together than non-platooning vehicles, and the vehicles will therefore take up less space on the roads. This could be necessary in the future as the population and the number of vehicles increase further. Another benefit will be fewer traffic jams, leading to decreased travel time, lower local noise levels and a further decrease in fuel consumption. Additional wear on the roads could become an issue if platooning becomes commonly used, because the platoons will be driving in the same tire tracks. This could lead to a higher amount of harmful particles in the air leading to a lower air quality. One way of addressing this could be to program the platoons of vehicles such that different platoons drive on different parts of each lane in order to evenly distribute the wear on the roads.

As the usage of platoons becomes more widespread, the cost of freight transportation with trucks will be reduced. This could lead to a shift away from non-road bound transportation such as air and sea bound transportation. A decrease in air freight would have an environmentally positive effect, while a decrease in sea freight would be negative. This is due to air transportation being generally less fuel-efficient than road bound transportation while sea transport is the most efficient mean of transport. However, this shift, as well as the lower costs, could lead to an increase of traffic on the roads to meet the demands of the consumer. With the same reasoning, fewer traffic jams could lead to an increased use of personal vehicles, which also leads to increased traffic on the roads.

Automation of vehicles can be implemented at different levels of autonomy. These vary gradually from manually driven to fully autonomous vehicles. Low levels of autonomy, such as adaptive cruise control and lane departure warning systems, would relieve drivers. However, high levels of autonomy can have a negative impact, since drivers still need to pay some attention but not enough to feel activated and stimulated on the job. In the case of platooning, stress can be caused from driving closely together without the ability to intervene.

Autonomous driving introduces multiple new ethical questions that need to be considered when developing autonomous driving. Much like the well-known “Trolley Problem”, an autonomous vehicle would have to be equipped to handle certain priorities and trade-offs in case of accidents. In difficult edge-case scenarios, a system could have either a utilitarian or a deontological approach. If an utilitarian approach would be adapted to accidents of autonomous vehicles, ethically complex prioritizations need to be implemented to minimize the consequences on lost lives or serious injuries. These could for example include considerations about the amount of people that could get injured, how severely the injuries would be, their age, social and financial position etc. Categorizing the value of people in such a way could be problematic. Deontologists might argue that doing nothing would be favorable since it is morally wrong to actively cause harm, even though it might cause less harm overall. In practice, this would mean having no control systems built into the platooning vehicles to prevent accidents.

Evidently, there are a lot of areas that still need to be studied and questions that need to be answered before autonomous vehicles and platooning can be a part of our daily life. However, if these issues can be solved, the advantages will exceed the disadvantages. Platooning will then have a great impact on the society and environment for current and future generations.

Model Predictive Control for Vision-Based Platooning Utilizing Road Topography

Mattias Hansson and Sofia Magnusson

***Abstract*—Platooning is when vehicles are driving close after each other at a set distance and it is a promising method to improve the traffic of todays infrastructure. Several approaches for platooning can be taken and in this paper a vision-based implementation has been studied. With a camera that detects the orientation of a marker attached to a small vehicle, it has been examined how the pitch of the marker can be exploited to perform vision-based platooning considering the road grade. A model predictive control strategy is presented to maintain a platooning distance with the potential of utilizing road topography. The aim of the project was to use this information to minimize brake and motor forces of the platooning vehicle. The strategy was based on relative vehicle states, detectable by a camera. The model predictive controller was implemented on small robotic vehicles and tested on a flat surface. The controller was successful in converging towards the wanted distance and capable of reaching a steady state speed. The results showed that it took 15 seconds for the system to reach a steady state.**

***Sammanfattning*—Konvojkörning är när fordon kör nära efter varandra med ett bestämt avstånd och det är en lovande metod för att förbättra trafiken i dagens infrastruktur. Åtskilliga tillvägagångssätt kan tas och i denna artikel så har ett visionsbaserat genomförande studerats. Med en kamera som upptäcker orienteringen av en markör som sitter på ett litet fordon så har det undersökts hur markörens lutningsvinkel kan utnyttjas för att utföra en visionsbaserad konvojkörning med hänsyn till vägens lutning. En model predictive control-strategi är presenterad för att bibehålla ett bestämt konvojavstånd med möjligheten att använda vägens topografi. Projektets mål var att använda denna information för att minska broms- och motorkrafter för det konvojkörande fordonet. Strategin grundades på fordonets relativa tillstånd som var detekterbara med en kamera. En model predicitve control utfärdades på små robotfordon och testades på en platt yta. Kontrollern var framgångsrik i att konvergera mot det önskade avståndet och kapabel till att nå ett stabilt tillstånd för hastigheten. Resultaten visade att det tog 15 sekunder för fordonets hastighet att nå det stabila tillståndet.**

***Index Terms*—ArUco marker, MPC, model predictive control, platooning, road topography**

Supervisor: Frank Jiang

TRITA number: TRITA-EECS-EX-2021:137

I. Introduction

Platooning is the concept of vehicles following one another while keeping a desired intervehicular spacing. This area of research has been studied for many years, and even though it has presented many positive results, there is still investigation that has to be done in order to implement platooning in the infrastructure.

The enabling of platooning in the near future is important as it would make a difference in many fields that today are negatively influenced by the effects of heavy traffic. Due to a reduction in drag, platooning can contribute to lower CO_2 emissions and hence reduce the impact on the environment [1] while also saving fuel by driving at consistent speeds [2]. This is beneficial from an economical point of view, as the cost of fuel is the second largest cost after personnel [2] for transport with heavy duty vehicles. The efficiency of driving in an organized platoon ensues a shorter travel time and is competent in handling higher traffic by keeping to the intervehicular spacing [1]. The partial automation of the trucks in a platoon adds to a safer driving environment. There have been projects that explore the health and safety of the driver, and how much the driver needs to be involved in controlling the vehicle when it is already in a platoon to ensure a well-conditioned driving experience [1].

A platoon can broadcast information among the vehicles in different ways, mainly by using communication or different kinds of sensors. The communication in a platoon is usually performed by connecting the vehicles to each other in one way or another. This way, the vehicles in the platoon can exchange signals or information to execute the same steering signals at the same time and thus platoon effectively. Platooning with sensors provide information about the vehicle in front of a platooning vehicle, for example by using radars or cameras. When using cameras a platoon becomes vision-based and can utilize a lot of information about the vehicle's surroundings. This has the advantage of not being heavy on precollected data, as it recieves all the necessary information in real time from what the cameras are detecting. Interacting with vision would also allow for vehicles from different manufacturers to collaborate [1], as the vision-based strategies are located in every single vehicle and does not depend on having a common framework.

In this project, the utilization of road topography with vision-based platooning was examined. It has been shown that heavy duty vehicles (HDVs) can benefit from using data about road topography for path planning to achieve a reduction in fuel consumption by look-ahead control (LAC) [3]. By combining look-ahead control with vehicle platooning, a potential for further fuel reduction has been shown in [4] by minimizing brake usage and maintaining a small platooning distance. One approach to this has been presented in [5] by calculating a fuel optimal speed profile for multiple vehicles in a platoon and using model predictive control (MPC) to follow the calculated speed profile. These methods rely on collected data about the road grade, which might not always be readily

available.

In this paper, a method for platooning while considering the road topography without relying on previously collected data is presented. A method of vision-based road grade estimation is shown to detect upcoming changes in road topography by visual tracking of a leader vehicle. A model predictive controller was designed to perform vision-based platooning with the objective of minimizing brake and motor forces and maintain a set distance. The performance of the formulated method was examined and it was tested if the model predictive controller could maintain wanted platooning qualities while also achieving a reduction in brake and motor usage.

II. Method

The method in this paper was designed to only rely on information available on the follower vehicle along with intervehicular tracking by ArUco marker detection with a camera. No other communication between the vehicles has been assumed. An MPC was formulated to perform platooning at a wanted distance while also accounting for the gradient of the road between the follower and leader vehicle.

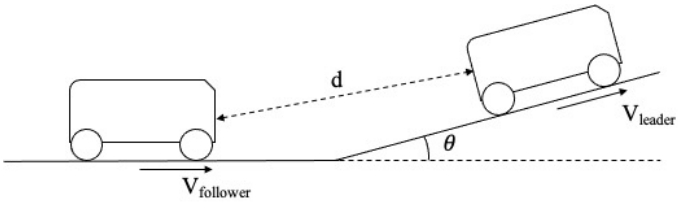


Fig. 1. Platooning vehicle following a leader before an uphill road segment.

In Figure 1, a vehicle is depicted driving on a flat surface while following a leader vehicle driving uphill on a road of angle θ with the intervehicular distance d. The follower vehicle relies on detecting a marker on the back of the leader vehicle with the help of a front mounted camera. By collecting information from the camera, the vehicle can detect the velocity, distance and pitch relative to the preceding vehicle by ArUco marker detection. This information is utilized by a model predictive controller with the objective of maintaining a wanted platooning distance while minimizing motor acceleration and brake deceleration. The MPC fomulation relies on real time road grade mapping by combining the marker detection of the leader vehicle with the road grade estimation of the follower vehicle. The method for the marker detection and the road grade estimation is presented in Section III. The MPC formulation is described in Section IV.

The MPC was then implemented on the Small Vehicles for Autonomy (SVEA) platform with the Robotics Operating System (ROS) and Python. Its performance was then evaluated with experiments on flat terrain.

Figure 2 depicts the vehicles on which the MPC was implemented. It shows how the SVEAs were platooning in the experiments, which were conducted with a 5x5 ArUco marker of size 100 mm on the back of the leader vehicle. The follower vehicle used a 1080p camera for detecting the marker ahead of it. The results of the experiments is presented in Section V. Finally, the model predictive controller's performance is evaluated and further research is discussed in Section VI.

Fig. 2. SVEAs platooning by ArUco marker detection in the Q building at KTH Campus.

III. Marker detection

In this project, ArUco markers have been used for vehicle tracking. With an ArUco marker attached to the rear of the leader vehicle, the camera on the follower vehicle can estimate the position and orientation of the marker. The markers have been utilized for identifying the distance to the preceding vehicle, estimating the vehicles relative velocity and detecting the pitch of the leader. Similar information can be gathered by other means, as shown in [6], where vision-based platooning is demonstrated by tracking a poster attached to the lead vehicle.

A. Relative velocity detection

The distance d from the camera to the marker gives the spacing between the vehicles. In this project, the distance has been estimated in the z-direction straight out of the ArUco marker. The relative velocity v_{rel} of the vehicles can then be estimated by the difference in distance

$$v_{rel} = \frac{d_{i-1} - d_i}{\Delta t_{sampling}}, \tag{1}$$

where d_i denotes the distance at a discretized sampling time t_i for the ArUco marker detection and $\Delta t_{sampling} = t_i - t_{i-1}$ is the time between the two samples. The relative velocity refers to the velocity of the marker detected by the camera. In this paper, a positive relative velocity has been defined for a decrease in distance to the marker. This definition comes from the formulation of the model, as mentioned in Section IV.

B. Road grade detection

ArUco markers also provide information about orientation. For this project, only the pitch of the marker is needed and

is used for the road grade detection. The pitch of the marker relative to the camera is denoted by θ. If the leader vehicle encounters a hill, the difference in pitch of the vehicle is detected by the camera from the marker's orientation. To obtain the absolute pitch of the leader vehicle and therefore the road grade, the pitch of the follower also has to be known. In this paper, an inertial measurement unit (IMU) has been used. An IMU can estimate the angle of the gravitational acceleration to obtain the orientation of the vehicle. However, as discussed in [7], there are several other ways to do this estimation. The pitch of the follower, as obtained from the IMU, is denoted by ϕ. Combining IMU data with the detected pitch of the leader, the actual road grade at the leader's position can be calculated as

$$\alpha = \phi + \theta, \tag{2}$$

where α is the pitch of the leader vehicle, and thus the road grade at its current position. The follower vehicle's position on the road is indicated with s, therefore the position of the leader vehicle is $s + d$. The angle α is then the road grade in position $s + d$, which is where the leader vehicle is currently situated. This means that the road grade can be modelled as a function of the position on the road, $\alpha(s)$, up to the current position of the leader vehicle.

IV. Model Predictive Control Strategy

The focus of this project is to minimize the brake and motor forces of the follower vehicle. This is shown in [4] to result in a lower energy consumption due to preservation of kinetic energy, with the largest potential for energy reduction coming from reduced brake usage. In order to find an optimal acceleration for the follower vehicle that takes this into account, an MPC has been implemented.

A. Physical model

The vehicle model is derived from Newton's second law of physics from the forces acting on a vehicle driving in the forward direction on a road of an arbitrary angle. The modelling is done in a similar fashion to [5], but with neglected air resistance and an adaptation to relative units.

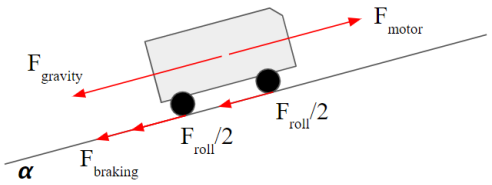


Fig. 3. Forces acting on a vehicle driving uphill.

The forces that are acting on the vehicle are presented in Figure 3, assuming that the system is driving with a positive velocity. The force from air resistance has been neglected in the model, because of low velocities and small frontal area of the SVEAs. In accordance with the forces in Figure 3, Newton's second law gives the acceleration of the vehicle,

$$ma = F_{motor} - F_{braking} - F_{roll} - F_{gravity} \tag{3}$$

where F_{motor} is the driving force from the motor, $F_{braking}$ is the brake force, F_{roll} derives from the rolling resistance between the tires and is proportional to the normal force acting on the vehicle and the road, and $F_{gravity}$ is the tangential component of the gravitational force parallel to the road.

The forces $F_{gravity}$ and F_{roll} depend on the angle of the road α that the vehicle is driving on, according to

$$F_{gravity} = mgsin(\alpha), \quad F_{roll} = c_r mgcos(\alpha), \tag{4}$$

where m is the mass of the vehicle, g the gravitational acceleration and c_r is the rolling resistance coefficient between the vehicle tires and the ground.

The model states are the intervehicular distance d and the relative velocity between the vehicles v_{rel}. The relative velocity is defined by

$$v_{rel}(t) = v(t) - v_{leader}(t), \tag{5}$$

where $v(t)$ is the velocity of the follower. The relation between the distance d and relative velocity v_{rel} is given by

$$\dot{d}(t) = -v_{rel}(t), \tag{6}$$

in which the negative sign is in accordance with the relative states used for the model and coincide with the visual detection presented in Section III.

B. Discretization

To implement the MPC, a discretized model is needed. In this paper, the time discretization is denoted by t_k where the difference between timesteps Δt correspond to the computation time of the MPC. A discretization of equation (3) gives

$$m\frac{v(t_{k+1}) - v(t_k)}{\Delta t} = F_m(t_k) - F_b(t_k) - F_g(t_k) - F_m(t_k), \tag{7}$$

where the notation for the different forces have been shortened to their first letters. Equation (7) can in turn be changed to depend on the relative velocity by substituting the velocity according to equation (5). This works under the assumption that the leading vehicle is driving at a constant velocity over the control horizon of the MPC,

$$v_{rel}(t_k) = v(t_k) - v_{leader}(t_1) \tag{8}$$

for $k \in \{1, ..., N_p\}$, where N_p denotes the number of samples in the prediction horizon. This indicates that the velocity of the leader vehicle is constant at every step that the MPC is solving the optimization problem, but not throughout the whole experiment. This assumption has to be made because the velocity of the preceding vehicle is non-controllable by the MPC for this vision-based approach. But because of the receding horizon fashion of an MPC, changes in velocity of the leader are still accounted for between MPC calculations.

By utilizing equation (8), the model in equation (7) can be turned into a relative one,

$$\frac{v_{rel,k+1} - v_{rel,k}}{\Delta t} = \frac{F_{m,k}}{m} - \frac{F_{b,k}}{m} - \frac{F_{g,k}}{m} - \frac{F_{r,k}}{m} \quad (9)$$

where a more compact notation has been used for the discretization by discarding the notation for the time t. In order to formulate the state-space representation, equation (6) is discretized by

$$\frac{d_{k+1} - d_k}{\Delta t} = -v_{rel,k}, \quad (10)$$

where d_k represents the distance between the vehicles and d_{k+1} is the predicted distance at the next sample of the prediction horizon.

For the complete state-space representation, a model on the form

$$\bar{x}_{k+1} = A\bar{x}_k + B\bar{u}_k + C_k \quad (11)$$

is required. In this model, $\bar{x}_{k+1}$, $\bar{x}_k$, $\bar{u}_k$, and C_k are vectors and A and B are matrices. The vector $\bar{x}_k = [v_{rel,k}, d_k]^T$ contains the vehicle states, given by the relative velocity and distance. The output vector $\bar{u}_k = [f_{m,k}, \quad f_{b,k}]^T$ gives the system outputs as motor acceleration and brake deceleration. With the discrete states in equations (9) and (10), along with the outside forces according to equation (4), the complete state-space representation of the system is given by

$$\begin{bmatrix} v_{rel,k+1} \\ d_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\Delta t & 1 \end{bmatrix} \begin{bmatrix} v_{rel,k} \\ d_k \end{bmatrix} + \quad (12)$$

$$\begin{bmatrix} \Delta t & -\Delta t \\ 0 & 0 \end{bmatrix} \begin{bmatrix} f_{m,k} \\ f_{b,k} \end{bmatrix} + \begin{bmatrix} -\Delta t(c_r g cos(\alpha(s_k)) + g sin(\alpha(s_k))) \\ 0 \end{bmatrix},$$

where the matrices are

$$A = \begin{bmatrix} 1 & 0 \\ -\Delta t & 1 \end{bmatrix}, \quad B = \begin{bmatrix} \Delta t & -\Delta t \\ 0 & 0 \end{bmatrix}. \quad (13)$$

In this formulation, the motor and brake forces are modelled as accelerations, hence the notations $f_{m,k} = F_{m,k}/m$ and $f_{b,k} = F_{b,k}/m$, that are used in a similar manner as in [5]. This way, the control signals that the MPC sends are not what forces to directly apply on the brake and motor, but the accelerations that correspond to the brake and motor forces applied by the vehicle.

The vector

$$C_k = \begin{bmatrix} -\Delta t(c_r g cos(\alpha(s_k)) + g sin(\alpha(s_k))) \\ 0 \end{bmatrix} \quad (14)$$

depends on the road grade at the position of timestep k.

C. Constraints

Model constraints have been set only on the control parameters of motor acceleration and brake deceleration,

$$0 \leq f_{m,k} \leq c_1 0.3 \quad (15)$$

$$0 \leq f_{b,k} \leq c_2 1.0, \quad (16)$$

where constants c_1 and c_2 are linear scaling factors. This can be done due to the high level interface of the SVEA, where the acceleration of the vehicle is directly controllable by changing the wanted speed of the platooning vehicle with the calculated acceleration $a(t_k) = f_{motor}(t_k) - f_{braking}(t_k)$.

Due to the relative nature of the model, the speed of the vehicle is constrained seperately. The speed of the vehicle is bounded by

$$0 \leq v(t_k) \leq v_{max}, \quad (17)$$

where v_{max} is a chosen speed limit. The velocity is bounded by the lower limit of zero to ensure no reversing is induced, as the vehicle model is accurate only for the case of driving in the forward direction of the vehicle and reversing is unwanted behaviour for platooning.

Safety is ensured with a separate controller, which initiates an emergency brake if the intervehicular distance becomes too small. Braking is then ensued if the spacing becomes less than the set safe distance,

$$d(t_k) \leq d_{min}, \quad (18)$$

where d_{min} is the minimum distance allowed, which has to be a distance where the vehicle can safely come to a halt without colliding into the vehicle in front.

D. Cost function

The cost function for the MPC ensures platoon maintenance while also minimizing unnecessary accelerations. The cost function is defined as

$$J = min. \sum_{k=1}^{N_p} ((\overline{x}_k - \overline{x}_{ref})^T Q(\overline{x}_k - \overline{x}_{ref})) + \sum_{k=1}^{N_p} (\overline{u}_k^T R \overline{u}_k), \quad (19)$$

where $\overline{x}_{ref} = [v_{rel,ref}, \quad d_{ref}]^T$ denotes the reference states and

$$Q = \begin{bmatrix} q_v, 0 \\ 0, q_d \end{bmatrix}, \quad R = \begin{bmatrix} q_m, 0 \\ 0, q_b \end{bmatrix} \quad (20)$$

are weight matrices. The reference for the distance d_{ref} indicates the wanted platooning distance, while the reference value for the relative velocity has been chosen as $v_{rel,ref} = 0$ m/s so that the vehicles maintain the same velocity when the desired platooning distance has been reached. The second term in equation (19) ensures minimization of control accelerations over the prediction horizon.

The weights in equation (20) decides the performance and what parts should be taken more into consideration by the

MPC. In this model, weights can be distributed on the relative velocity with q_v, the intervehicular distance with q_d, the motor acceleration with q_m, and the brake deceleration with q_b. Due to the goal being mainly to minimze the motor and brake forces, larger weights were put on q_m and q_b. The exact tuning of the weights were decided by experimentation and is discussed in Section V.

E. Model predictive controller

The complete MPC formulation is as follows,

$$\begin{aligned} &min. \quad J(\bar{x}_{ref}, \bar{x}_k, \bar{u}_k) \\ &subj. \quad to \quad \bar{x}_{k+1} = A\bar{x}_k + B\bar{u}_k + C_k \\ &\qquad 0 \leq f_{m,k} \leq c_1 0.3 \\ &\qquad 0 \leq f_{b,k} \leq c_2 1.0, \end{aligned} \tag{21}$$

where the objective is to minimize the cost function in equation (19) subject to the state-space representation in equation (12) and the constraints in equations (15) and (16).

The MPC is based on convex quadratic optimization with linear constraints. The problem can be solved in a short time frame with solvers such as OSQP [8], which is the solver used in this paper. Based on the dimension of the problem it is solvable in a small fraction of a second.

V. Results

The MPC was implemented on the SVEA and experiments were performed on two vehicles, one follower and one leader. The vehicles started at a stand still after which the leader was accelerated to and held a constant speed. The follower vehicle was controlled by the MPC by using ArUco marker detection.

A. Road grade evaluation

The goal of the road grade evaluation is to successfully estimate the angle of the driven road. This was tested on a flat surface with the MPC controlling the follower vehicle.

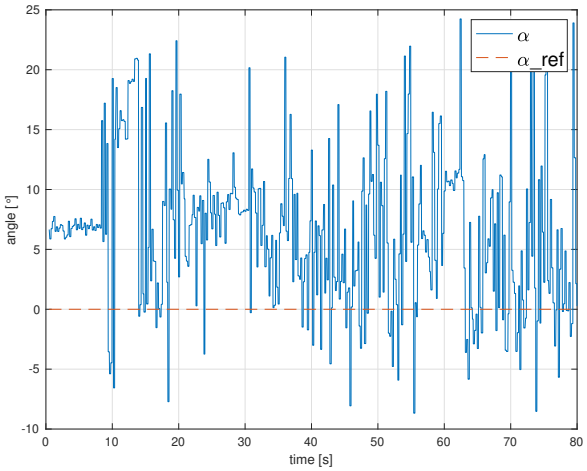


Fig. 4. Estimated leader vehicle pitch while driving on a flat surface.

As seen in Figure 4 the road grade evaluation did not correctly estimate the actual road grade of 0°. The vehicles started at a stand still and started moving after approximately 10 seconds.

B. Model predictive controller

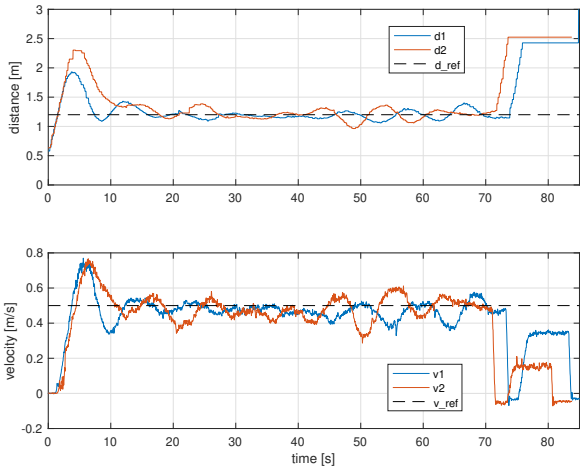


Fig. 5. Distance and velocity graphs from experimentation on SVEA obtained from two different tests with the same MPC parameters. Index "1" indicates the first test and index "2" indicates the second one.

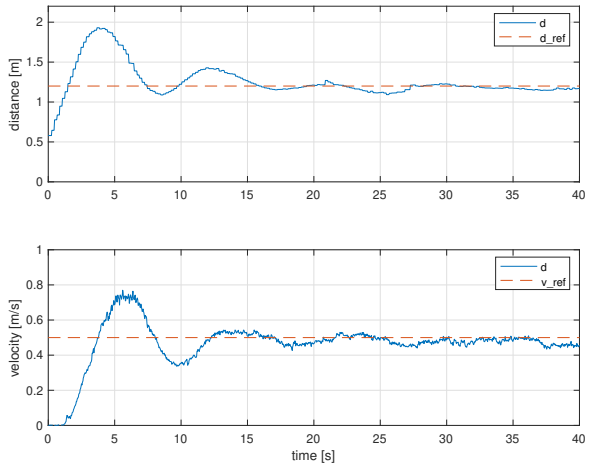


Fig. 6. Distance and velocity graphs from experimentation on SVEA representing the first half of test 1.

Figure 5 shows the MPC converging towards the steady state at first and then how disturbances occur until the MPC unexpectedly brakes towards a halt at approximately 70 seconds. Soon after this, the vision of the marker is lost, as seen by the constant observed distance between the two vehicles, which corresponds to the last picked up distance by the camera. The tests were made with the same parameters and with as similar experimentation setup as possible. The experiments were done with MPC weights $q_v = 5$, $q_d = 3$, $q_m = 20$, and $q_b = 20$. To

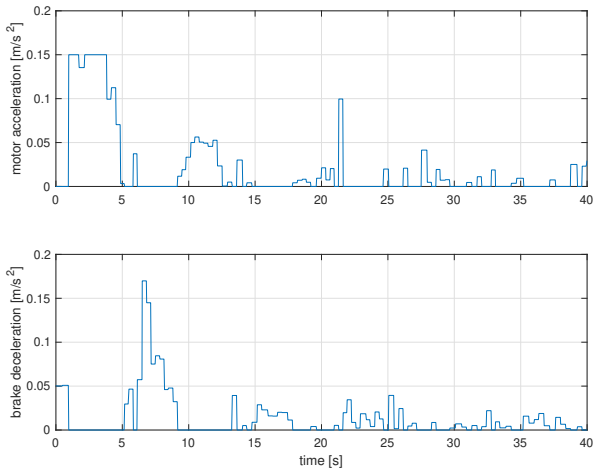


Fig. 7. Motor acceleration and brake deceleration from experimentation on SVEA representing the first half of test 1.

focus on the step response, Figure 6 and Figure 7 shows the first half of the first experiment from Figure 5.

The experiments were conducted with both vehicles starting at a stand still after which the leader quickly accelerated to 0.5 m/s, which was held constant for the remainder of the experiment. The motor acceleration was bounded by the upper constraint of 0.15 m/s^2 and the input velocity of the follower was constrained by a speed limit of 0.7 m/s. These tests were conducted on flat surface and the road grade evaluation was not fed into the MPC.

In Figure 6, it is seen how the distance between the two vehicles is converging towards the desired intervehicular spacing at 1.2 m after 15 seconds. A similar result is shown in the graph representing the velocity in Figure 6, where a steady state velocity of 0.5 m/s is reached after 12 seconds. The graph also shows how the velocity increases when the vehicle detects a distance further than the reference. The references for the velocity and the distance are marked with the dotted lines.

As seen in the graph of the brake deceleration in Figure 7, the SVEA is braking at the start of the experiment. As the brake deceleration goes down, a matching increase in motor acceleration is observed in its corresponding graph in Figure 7. This pattern continues throughout the graphs, demonstrating the relation between the brake and motor forces and indicating how the vehicle is compensating for the increased distance or velocity with the accelerations and converging towards a steady state. When the steady state is reached, brake and motor accelerations are small except for some deviations, mainly a spike in motor acceleration at $t = 22$ seconds. This spike corresponds to the disturbance seen in the distance graph at the same time.

VI. Discussion

The road grade evaluation did not meet the desired qualities. While driving the vehicles on flat surface by assuming a road grade of 0° for the MPC, the performance of the platoon was satisfactory in the conducted experiment and a steady state was reached 15 seconds after the start of the experiment. The visual tracking was unreliable, which disrupted the performance of the MPC.

A. Road grade evaluation

The experiments were conducted on flat terrrain, so the evaluation of the road grade was expected to be around 0°. However, the results showed that the detected inclination of the road did not correspond to the actual one. There are several factors that may have contributed to this result, where the most plausible one could be the setup of the experiment. The ArUco marker was attached to the leader vehicle with a simple piece of cardboard and adhesive tape. The camera was also attached with tape to the front of the follower vehicle. From early experiments, the detection of the pitch had been tested while holding the marker in front of the camera. It showed accurate results for pitch and distance even for small movements of the marker, so the camera detection is sensitive for deviations. This can be seen in the first 10 seconds of Figure 4. The vehicles are still idle and so the observed pitch of the leader is approximately 6°. Even though this is not 0°, it could still be compensated for as it might depend on the placement of the camera and the marker. Then the vehicles are set in motion and the detection becomes inaccurate. This is likely due to the setup of the camera and the marker being shaky when the SVEAs are moving. Also affecting the results are the large suspensions of the SVEA vehicles. Because of time constraints on the project, this problem could unfortunately not be resolved and therefore no experiments were conducted on slopes. For the experiments, the MPC was given a constant road grade of 0° instead of using this data.

B. MPC weights

The experiments favoured a larger MPC weight for the relative velocity q_v than the distance q_d. With higher weights on the distance, the overshoot was too large and emergency braking or loss of marker vision was ensued. By putting more weight on the relative velocity, the overshoot wasn't as large because the controller favoured minimizing the relative velocity between the vehicles, resulting in a more stable system. Due to the implementation in Python, the MPC had a long compilation time and could only perform calculations about four times per second. This might have affected the result in many ways, but had been taken into account by performing experiments at low velocities. The acceleration was constrained by the factor $c_1 = 0.5$ from equation (15) to limit too aggressive acceleration. Limiting the motor acceleration decreases overshooting at the cost of a slower rise time. The scaling factor for the brake acceleration was set to $c_2 = 1.0$. The MPC weight for braking q_b was chosen as a large value because of the potential that brake reduction could be the largest energy saving strategy from conservation of kinetic energy [4]. The weight of the motor q_m was chosen as the same value as for the brake. Increased motor and brake weights also led to a more stable system, which was deemed favourable.

C. Solving time and data processing

The camera was set to 20 fps and the ArUco marker detection was run with a frequency of 10 Hz. This was set due to the heavy amounts of processing used up by the MPC and different processes running on the SVEA. A camera with a higher fps would show an improvement in the detection of the ArUco marker and faster and more accurate readings of both the distance and the relative velocity. When increasing the frequency of the marker detection, too much processing was used up and the MPC got slower. The trade off between accurate detection and MPC solving time could be avoided by implementing the same strategy in C++, as it is a faster programming language. Then the usage of a higher frequency for the marker detection would not impair the solving time of the MPC. Because of slow computation in the implementation, a short prediction horizon of $N_p = 6$ was selected. Even though the prediction horizon is quite short, it results in a time horizon of 1.5 seconds given the computation time of 0.25 seconds.

D. Unreliability

One substantial flaw with this method is the unreliability of the marker detection. As seen in Figure 5, both experiments suddenly start oscillating after having reached a steady state. This is most likely due to the relative velocity that's being fed into the MPC. As seen in the distance graph in Figure 5, the distance tracking is reliable up to a distance of about 2.5 m. After this point the distance in the graph becomes flat, represented by the latest distance picked up by the camera before loss of marker vision occurred. A protocol was developed where at loss of marker vision the MPC was sent an acceleration of 0 m/s^2, which can be seen as the constant velocities between 70 and 83 seconds for the two tests. The tests were then interrupted, representing the loss down to 0 m/s for the follower.

This is intended behaviour while the first sudden spikes down to 0 m/s in Figure 5 is unwanted behaviour, aswell as the oscillation preceeding this. The relative velocity fed into the MPC is therefore the most likely cause. The relative velocity is calculated according to equation (1). The ArUco marker detection was set to 10 Hz and small changes in the percieved distance then gives a large deviation in the percieved velocity. For example, a change in percieved distance of 0.01 m from the reference gives a percieved relative velocity of ± 0.1 m/s, which represents a deviation of 25% from the reference velocity of 0.4 m/s. If the MPC calculation coincides with an error this large, the MPC, running at approximately 4 Hz, can further propagate this error. This is likely the cause of the sudden oscillations and sudden halt seen in Figure 5.

E. Rolling force

The rolling force was neglected by setting the rolling constant $c_r = 0$ for the experiments. This was done due to the implementation on the SVEA, where the calculated control outputs represented by the brake and motor accelerations were implemented by changing the desired speed of the vehicle. If experimentation was done with $c_r > 0$, then the MPC would converge towards a smaller distance, which is unwanted behaviour. To account for the rolling resistance another implementation method would be needed. This meant that all the deceleration of the vehicle was calculated by the MPC and modelled as brake deceleration. Because the implementation of the MPC did not consider any natural deceleration, the brake and motor usage in Figure 7 are of the same magnitude. A brake reduction is only observed when the vehicle reaches a steady state by a total reduction in acceleration.

F. Future work

Further research on this method would be to collect road grade data from a slope and implement it with the MPC on the SVEA, to see how this affects the performance of the presented control strategy. Experiments could later be conducted with the SVEA or a similar platform on roads with changing road grade to test the real time grade estimation presented in this paper. This way, more experiments could be done to investigate the behaviour of the MPC and examining the achievable reduction in brake and motor forces. In case that the method proves to be advantageous for these conditions, a more efficient implementation could be developed for increased performance and faster MPC solving times. If the method seems promising, it could be implemented and tested on cars as a form of adaptive cruise control that takes road grade into account. The method would probably not prove too favorable for a heavy duty vehicle platooning, as the main focus in that field lies on minimizing the distance to the preceding vehicle, while the focus in this method lies on minimizing the control forces over the prediction horizon. In adaptive cruise control on the other hand, which is often done with greater intervehicular distances, the method presented might prove beneficial over other conventional methods. The method might also be adapted with more constraints or other states for further improved performance.

VII. Conclusions

In conclusion, the formulated MPC is capable of reaching a steady state by tracking a leader vehicle with ArUco marker detection under ideal conditions. The pitch detection strategy did not work for the conditions that the experiments were conducted in. The vehicle tracking and control strategy functioned for a vehicle platoon consisting of two SVEAs driving on a flat surface, but showed unreliable behaviour. The controller reduces brake and engine forces only when reaching a steady state, but no further reduction in brake usage from the strategy could be observed.

Appendix A

Code of the Model Predictive Controller.

Acknowledgment

The authors would like to thank their supervisor Frank, who has provided guidance and support throughout the whole project.

REFERENCES

[1] S. Sivanand and M. S. Gajanand, "Platooning for sustainable freight transportation: an adoptable practice in the near future?" *Transport Reviews*, pp. 581–606, 2020.

[2] F. Chen, X. Ma, X. Pan, and L. Zhang, "Economy in truck platooning: A literature overview and directions for future research," *Hindawi*, 2020.

[3] E. Hellström, "Look-ahead control of heavy trucks utilizing road topography," p. 80, Linköping University - The Institute of Technology, Linköping, 2007.

[4] L. Bühler, "Fuel-efficient platooning of heavy duty vehicles through road topography preview information," Master's thesis, KTH - Royal Institute of Technology, Stockholm, 2013.

[5] V. Turri, B. Besselink, J. Mårtensson, and K. H. Johansson, "Fuel-efficient heavy-duty vehicle platooning by look-ahead control," pp. 654–660, Dec. 2014.

[6] S. Benhimane, E. Malis, P. Rives, and J. Azinheira, "Vision-based control for car platooning using homography decomposition," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 2161–2166.

[7] P. Sahlholm, "Distributed road grade estimation for heavy duty vehicles," Ph.D. dissertation, KTH - Royal Institute of Technology, Stockholm, 2011.

[8] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: an operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020. [Online]. Available: https://doi.org/10.1007/s12532-020-00179-2

Stability of a Vision Based Platooning System

Kristina Kjellberg and Ann Köling

***Abstract*—The current development of autonomous vehicles allow for several new applications to form and evolve. One of these are platooning, where several vehicles drive closely together with automatic car following. The method of getting information about the other vehicles in a platoon can vary. One of these methods is using visual information from a camera. Having a camera on-board an autonomous vehicle has further potential, for example for recognition of objects in the vehicle's surroundings. This bachelor thesis uses small RC vehicles to test an example of a vision based platooning system. The system is then evaluated using a step response, from which the stability of the system is analyzed. Additionally, a previously developed communication based platooning system was tested in the same way and it's stability compared. The main conclusion of this thesis is that it is feasible to use a camera, ArUco marker and an Optimal Velocity Relative Velocity model to achieve a vision based platoon on a small set of RC vehicles.**

***Sammanfattning*—Forskningsframsteg inom området autonoma fordon möjliggör utveckling av ett flertal nya tillämpningar. En av dessa är platooning, som innebär att flera fordon kör nära varandra med automatisk farthållning. Metoden för att erhålla information om de andra fordonen i platoonen kan variera. En av dessa metoder är att använda visuell information från en kamera. Att ha en kamera ombord på ett autonomt fordon har stor potential, exempelvis för detektering av objekt i fordonets omgivning. Det här kandidatexamensarbetet använder små radiostyrda bilar för att testa ett exempel av ett kamerabaserat platooning-system. Systemet är sedan utvärderat med hjälp av ett stegsvar, från vilket stabiliteten av systemet är analyserat. Dessutom testas ett tidigare utvecklat kommunikationsbaserat platooning-system, hittills bara testat i simulering, på samma uppsättning bilar. Den huvudsakliga slutsatsen av detta arbete är att det är möjligt att använda en kamera, ArUco markör och en Optimal Velocity Relative Velocity modell för att uppnå en kamerabaserad platoon med en liten uppsättning radiostyrda bilar.**

***Index Terms*—Adaptive Cruise Control, Aruco markers, Real-time visual tracking, Vehicle platooning, Vision-based control**

Supervisor: Frank Jiang

TRITA number: TRITA-EECS-EX-2021:138

I. Introduction

Road transportation has accounted for 20% to 30% of greenhouse gas emissions in the European Union in the last two decades [1]. In efforts to lower this high percentage, researchrs and companies strive to develop solutions for fuel optimization. One of these solutions is platooning, a scenario where several vehicles, more or less autonomous, drive with greatly reduced longitudinal distances between them. The short distances lead to decreased air drag, in turn leading to reduced fuel consumption.

The development of autonomous vehicles is proceeding in steps, from non-automated to fully automated, with some driver assisting features in between. These steps apply for platooning as well, with features such as cruise control useful for platooning implementation.

A. Background

Driver assisting control systems are now more or less standard features is new cars. The most common system is Adaptive Cruise Control (ACC), using radar to gather information about the distance to and the velocity of the vehicle ahead [2]. A system adding to the ACC is the Cooperative Adaptive Cruise Control (CACC) using a communication system together with the sensor information. The CACC enables connection to several vehicles, for example in a platoon. However, both the ACC and the CACC can be used for platooning. Two models corresponding to these two systems are the Optimal Velocity Relative Velocity (OVRV) model and the Cooperative OVRV (C-OVRV) model, the first being quite common and the latter proposed in [2].

A lot of recent platooning related research is based on communication between vehicles to control the inter-vehicular spacings of the platoon. The communication opens up for new opportunities and can potentially lead to more stability. However, wireless communication for platooning demands a substantial amount of data being transferred. This could pose a problem if platooning solutions are scaled and more commonly used [3], why a fallback system using a non-communicative system would be a good option.

One, more simple, alternative solution to wireless communication is using only local information, from for example a radar, lidar or camera, to determine the distance to other vehicles in the platoon. The different types of sensors have different strengths, for example, radar is more robust while a camera can give far more semantic information [4], such as identification of arbitrary objects. With machine learning and computer vision technologies developing, having a camera mounted on a vehicle can therefore lead to greater possibilities.

Previous research has shown that vision based platooning can be implemented with, for example, an LQ (linear quadratic) model [5] or a modified pure pursuit algorithm [6]. However, not with an OVRV model. One advantage of using local sensor information together with an OVRV model is the fact that it would facilitate the transition to a C-OVRV model for communication. It would therefore be suitable for going back and fourth between communication and local information.

B. Project Formulation

This bachelor thesis aims to investigate if it is feasible to use a camera together with an OVRV model to achieve vision based platooning. Furthermore, how does the stability of a vision based platooning system compare to a cooperative

communication based one in a situation where the vehicles have to decrease their velocities? In regards to this, do the two types of signals seem to be compatible? If a communication based platoon would fail due to an excessive amount of data transferring, could a fallback system with vision be used? In this project, small RC vehicles equipped with cameras are used to investigate these questions.

C. Contributions

The contributions of the thesis project are to (1) platoon with RC vehicles using an OVRV model and a camera; (2) test the implementation of a C-OVRV model, presented by [2], using a small scale platoon of RC vehicles; (3) evaluate and compare the stability of the two systems and discuss the possibility of a fallback integration of them.

D. Thesis Outline

The thesis is structured as follows. In section II practical and theoretical information about the used systems is presented. This includes both software control models and hardware description. The testing conditions are presented in section III. In section IV, the results from the tests are given. The results are then discussed in section V, and finally the work is concluded in section VI.

II. Prerequisites

A. SVEA - Small Vehicles for Autonomy

The experiments in this thesis were executed on real systems. These systems consist of Small Vehicles for Autonomy (SVEAs), pictured in Figure 1. The SVEAs are developed at the Smart Mobility Lab at KTH Royal Institute of Technology, as an experimental platform for research about automated and connected vehicles. A SVEA consists of an RC vehicle equipped with a computer and several sensors (for example a lidar and a RealSense tracking camera). The SVEAs depend on ROS (Robotic Operating System) which allows the vehicles to communicate with each other through nodes and topics. For the implementation of vision based platooning in this project, another camera was mounted on the SVEA acting as a follower in the platoon. The video stream from the camera was published as a topic in the ROS network.

B. Fiducial Markers

In computer vision applications such as robot navigation, it's important to be able to estimate the pose and orientation of objects in the robot's field of view. A common solution for this is using binary fiducial markers, such as ArUco markers, together with a detection software. An example of an ArUco marker can be seen in Figure 2a.

In this project, ArUco markers were used to visually determine the distance between two SVEAs. The marker was placed on the rear bumper of the SVEA acting as a leader in the platoon. The video stream from the following SVEA's camera was subscribed by an ArUco detector, returning information about the ID and the position of the marker. The detection of a marker is visualized in Figure 2b.

SVEA

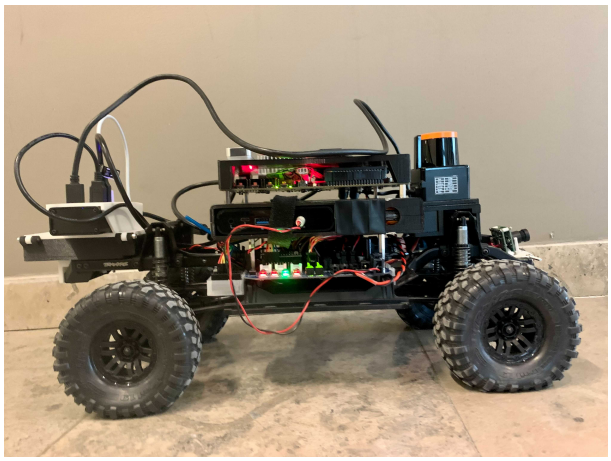


Figure 1. An example of a Small Vehicle for Autonomy system, used for the implementations in this thesis.

ArUco Markers

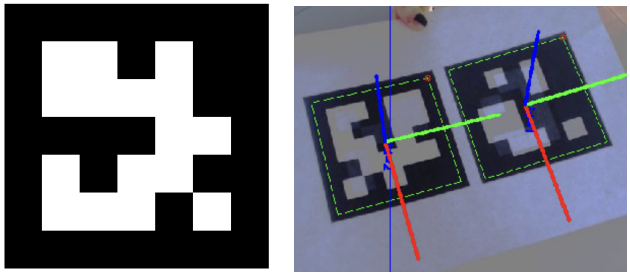


Figure 2. a) A 5x5 ArUco marker with ID 2, generated from [7]. b) Printed markers with ID 1 and 2, detected with an ArUco detector.

C. Optimal Velocity Relative Velocity Model

To control the speed of the platooning vehicles, a controller, originally developed for cruise control, was adapted. The model was modified and presented in [2] and gives the acceleration of a following vehicle as

$$\ddot{x} = k_1(s_i - \eta_i - \tau_i \dot{x}_i) + k_2 \dot{s}_i \quad , \tag{1}$$

where:

$\ddot{x}$ = computed acceleration of following car $[m/s^2]$
s_i = inter-vehicular spacing $[m]$
η_i = minimal inter-vehicular spacing allowed $[m]$
τ_i = desired time-headway $[s]$
$\dot{x}_i$ = follower velocity $[m/s]$
$\dot{s}_i$ = relative velocity $[m/s]$
k_1 = time-headway gain $[1/s^2]$
k_2 = relative velocity gain $[1/s]$

The acceleration is used to adjust the follower velocity and thereby maintain the desired inter-vehicular spacing of the platooning vehicles. The computed acceleration is dependent on the spacing between, and the velocity of, the follower and leader vehicles. The information about the leader is usually obtained by radar in ACC systems, making it non-communicative. However, the information could also be

obtained by other sensors or by communication. The solution that is used in section III assumes that spacing information is acquired from a sensor (camera) while the leader velocity is considered a *road standard*, meaning the platoon should aim for the speed limit when possible.

As seen in equation (1), the first term considers the distance between the vehicles and is scaled with k_1. The second term scales the relative velocity with k_2. These constants have to be tuned for a simulation or real system.

D. Cooperative Optimal Velocity Relative Velocity Model

A communication based version of the OVRV model takes in information of more vehicles in the platoon, than just the one directly ahead. An extension of (1), proposed in [2], gives the following equation:

$$\ddot{x}_i = k_1(s_i - \eta_i - \tau_i \dot{x}_i) + k_2 \dot{s}_i + w_i + k_3 \sum_{j \in N_i^+} (\hat{\dot{x}}_j - \dot{x}_i) + k_4 \sum_{j \in N_i^+} \left(\hat{x}_j - x_i - \hat{l}_j - \sum_{k=j+1} (\eta_k + l_k - \tau_k \hat{\dot{x}}_k) + l_i \right), \tag{2}$$

where the first two terms are the same as in (1). w_i is a disturbance, N_i is the number of vehicles that share their information with vehicle i, and the summations add the information of each of those vehicles.

As this project is not focused on the communicative system, but only uses it to compare the stability of the vision system, the equation is not explained in depth. For a thorough description, see [2].

E. String Stability

It is common that traffic jams appear without any obvious causes, such as accidents. Such scenarios are called phantom traffic jams and may occur when the traffic flow is string unstable. This means that the deceleration of a vehicle has amplified effects on the vehicles upstream the traffic flow [8].

To properly analyze the string stability of a platoon, an actual platoon of several vehicles is needed. However, the stability of each individual system could give an indication of how sensitive the platoon would be to sudden changes.

For example, an overshoot in the step response affects the platoon as the following vehicles would increase the overshoot as they would react to an overshoot instead of an ideal step response. It is established in [2] that a C-OVRV model can be used to decrease the overshoots in such a scenario and thereby create a string stable platoon. Even though it is not treated in this paper, it would be of interest to investigate how a vision based system would behave in terms of string stability.

F. Step Response

A common way to analyze a controller's stability and characteristics is through a step response. A step response means a reaction to an instant change (a step) as the system is in a steady state. How well the system follows the ideal step provides valuable information. Parameters that can be calculated to analyze the system are for example the overshoot and steady state error.

G. Fallback System

A fallback system is a backup for a main system and can be both hardware or software related. In this project, the fallback system refers to a backup for when the main platooning system using communication is not functioning, for example if data transferring is not working correctly. The backup system using vision would then be used instead. The fallback could be implemented the other way around as well. If a main system using vision is not working, for example if the view of the camera is obstructed, a backup system using communication could be used.

III. Implementation

The process of implementing the vision based platooning system for the SVEAs was made in four major steps. The first step was integrating a camera input to a ROS network. The following step was to process the video stream from the camera with an ArUco detection program giving the distance to visible ArUco markers. The next step was to modify and simplify an existing C-OVRV simulation to use the distance from the detector without communication with other vehicles. Finally, the code was adapted to run on the SVEAs. To monitor and analyze both simulation and SVEA runs, information from the ROS network was displayed in the visualizing program RVIZ.

The implementation and system structure can be described with the flow chart presented in Figure 3. The system is built on several scripts and ROS packages that receives, treats and transmits relevant information to and from the SVEAs.

This thesis builds upon the SVEA platform and to repeat the conducted experiments it is of value to have access to the KTH Smart Mobility Lab repository found in [9].

A. Simulation and Real System

A key part of this project was the understanding of the SVEA hardware and the implementation and modifications that are needed when real systems, not simulations, are used. Specifically, the model's gain (k_1 and k_2) had to be altered since the SVEAs are effected by more disturbances than what the simulation takes into account. Furthermore, the development and evaluation of vision based platooning was dependent on the use of a real system. This is because it requires the camera position and the inter-vehicular spacing to be changed simultaneously. Early in the project, some simulations were done together with real camera input. These simulations gave understanding and an indication of the function of the used model. However, no results were based on these simulations.

B. Experimental Setup

With the aim to compare and analyze the stability of an OVRV vision based platoon and a C-OVRV communication based one, the following two experiments were conducted.

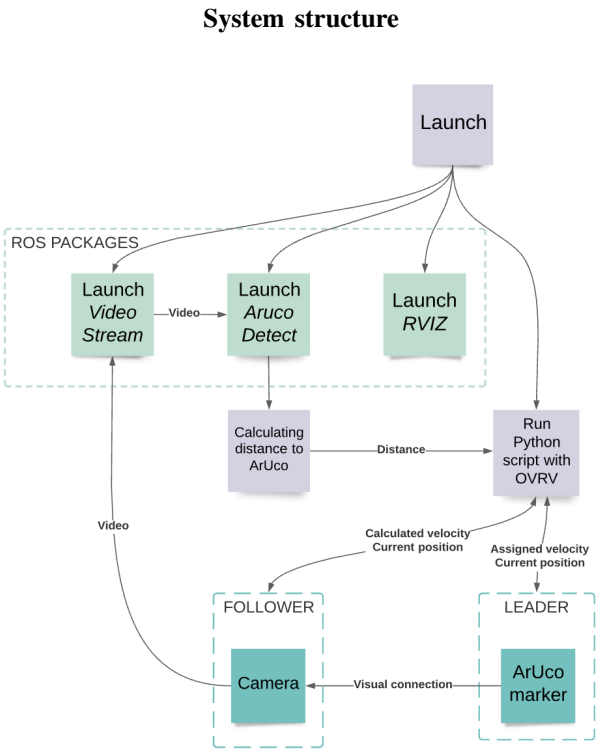


Figure 3. The system structure of the implemented programs demonstrating how information and instructions are transferred within the system.

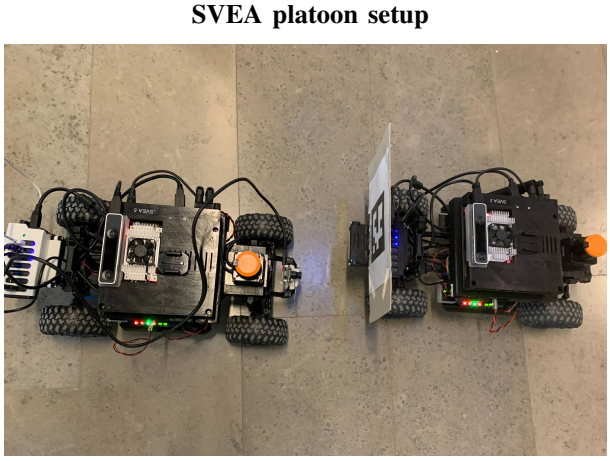


Figure 4. The setup of a two SVEA platoon for the first experiment. The left vehicle equipped with a camera and the right with an ArUco marker.

1) Experiment one: The first experiment was done to evaluate the characteristics of the step response using the developed vision based platooning system.

Two SVEAs were used, the first of the two with an ArUco marker mounted on its rear bumper and the second one equipped with a camera, this setup is pictured in Figure 4. The two vehicles were run with separate scripts following separate phantom leaders. The phantom leader is a simulated vehicle that gives the SVEA information about what velocity it should strive for and the path it should follow. The setup is depicted

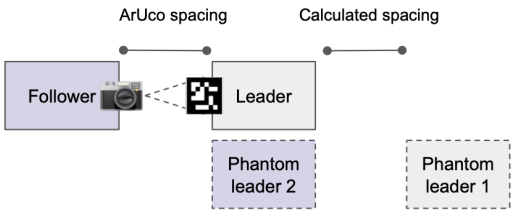


Figure 5. Block diagram visualizing the phantom leaders and the vehicles in the experimental setup for experiment one.

in Figure 5. The first of the two vehicles used a calculated spacing parameter between the position of the phantom leader and the vehicle itself. The second one used the visual distance to the ArUco marker.

Both SVEAs were started and the scripts were initiated. As the leader had reached a steady state, it was introduced to a disturbance velocity (a lower velocity representing a braking scenario) to get a step response of each of the two SVEAs.

2) Experiment two: The second experiment was conducted to evaluate the step response using an existing communication based script that adapts the C-OVRV model.

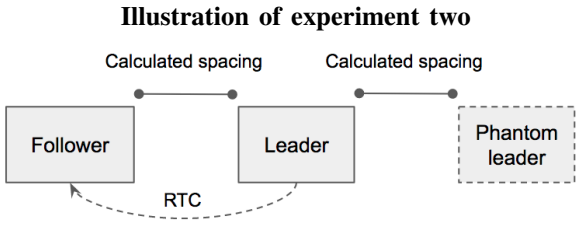


Figure 6. Block diagram visualizing the experimental setup for experiment two with two vehicles and one phantom.

Once again, two SVEAs were used, but this time they were connected using real-time communication (RTC) and only one phantom was used. Information about the leader and the phantom was sent from the leader to the follower using RTC. The setup is depicted in Figure 6. The two vehicles were still run with separate scripts, but now following the same trajectory and phantom leader. Both vehicles used calculated spacing parameters. Alike experiment one, the leader reached a steady state before the desired velocity was reduced to get a step response.

IV. Results

A. Experiment one

Based on Figure 7, the leader's and follower's overshoot can be calculated to 19.7% and 32.5% respectively. The steady state errors are determined to 0.07 m/s and 0.04 m/s for the leader and follower. These numbers are presented in Table 1.

The following vehicle never reached a steady state before the velocity step and neither of the vehicles reached steady state after the step. The lack of stability is problematic and made it difficult to analyze the data and get accurate values for the overshoot.

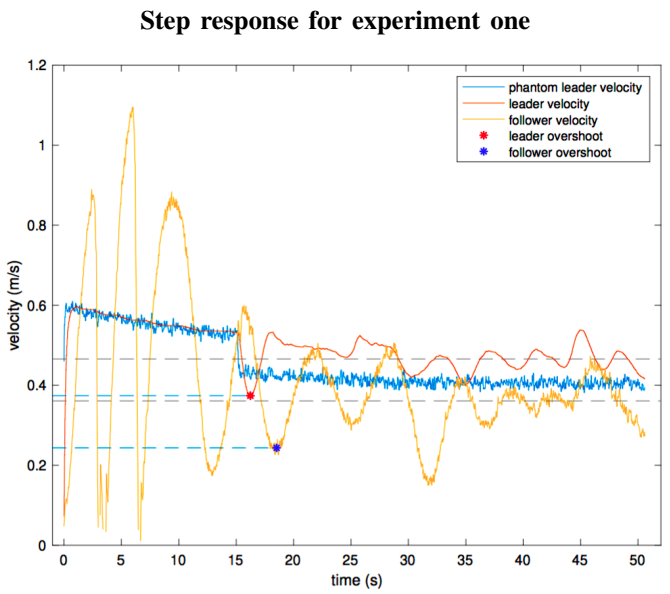


Figure 7. The figure demonstrates the velocities in respect to time in a platoon with two vehicles. The velocity of the first of the two phantoms is also graphed. The leader has an asymptotic velocity of 0.47 m/s and the follower has one of 0.36 m/s. After the step response, the amplitude of the leader's first peak is 0.37 m/s. Correspondingly, the amplitude of the follower's first peak is 0.24 m/s.

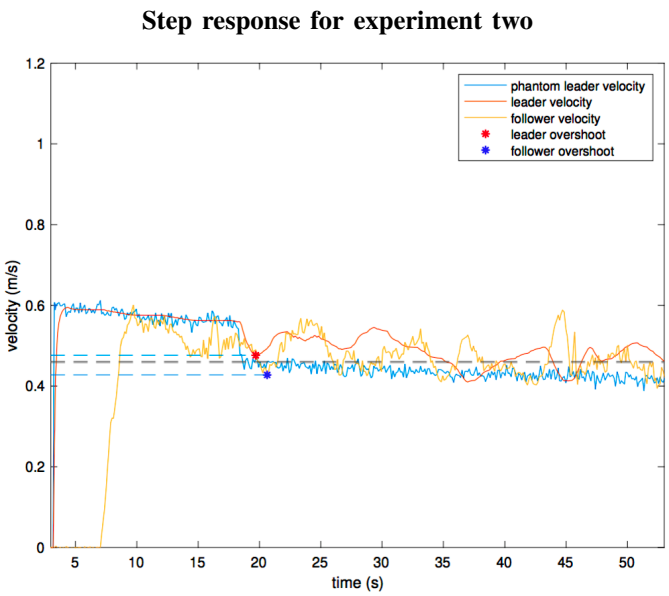


Figure 8. The figure demonstrates the velocities in respect to time in a platoon with two vehicles and one phantom leader. The leader has an asymptotic velocity of 0.462 m/s and the follower has one of 0.458 m/s. After the step response, the amplitude of the leader's first peak is 0.476 m/s. Correspondingly, the amplitude of the follower's first peak is 0.428 m/s.

Moreover, the vehicle velocities in the graph seem to be oscillating around two different asymptotes. As seen in Figure 7, the phantom's velocity after the step is 0.40 m/s, the leader's is 0.47 m/s and the follower's is 0.36 m/s. In reality this would mean that the distance between the two SVEAs was constantly increasing. However, this was not the case.

Another observation from Figure 7 is that the leader's velocity is oscillating and *differing from* the phantom's, but in turn the follower's is *varying with* the leader's. This indicates that the stability of the vision based system (which the follower uses) is better than the computing system (which the leader uses), at least when given sufficient time to stabilize.

B. Experiment two

The results from experiment two is computed from figure 8 and presented in Table 1. The steady state errors are 0.062 m/s for the leader and 0.058 m/s for the follower. The leaders undershoot is calculated to 3.3% and the followers to 6.7%.

Neither the follower or the leader reached steady state and there is no significant decrease in the oscillations. Furthermore, the leader reacts to the step with an undershoot rather than an overshoot, meaning that the leader's velocity is higher and not lower than it's asymptotic value.

Another observation from experiment two is that the leader and the follower have the same asymptotic values. This indicated that the follower's velocity is in line with the leader's which is crucial in a platoon. Additionally, their asymptotic velocity differs approximately 0.06 m/s from the phantom's. This constitutes 15% of the phantom leader's velocity (the desired velocity).

Table I
A TABLE COMPILING THE VALUES CALCULATED FROM THE FIGURES OF EXPERIMENT ONE AND TWO.

		Overshoot / Undershoot	Steady State Error
Exp 1	Leader	19.7%	0.07 m/s
	Follower	32.5%	0.04 m/s
Exp 2	Leader	3.3%	0.062 m/s
	Follower	6.7%	0.058 m/s

V. Discussion

The project outcome was in several ways not as expected. The difficulties of working with and running experiments on real systems, compared to a simulation, impeded the project and made it challenging to get the desired results. However, this was part of the project formulation: to investigate how and if the implementation on real vehicles works.

A. Potential for Development

One of the limiting factors for the experiment was the physical distance available. A longer runway would give more time for the systems to stabilize. It would be particularly desirable to increase the run time before the velocity step. Since the follower seems to be settling, this would increase the chances of the follower reaching a steady state.

It is also worth mentioning that the results presented in both experiment one and two were from single runs. In order to draw definite conclusions, more data would have to be collected from several runs. This is particularly important as the tests are performed on a real vehicles, were uncertain variables such as the SVEA's localization functionality, might effect the results.

When analyzing the stability of the system it is important to take into consideration the fact that the follower reacts to a step response rather than a step, and does thereby not have the ideal prerequisites. At the same time, this provides a realistic representation of the actual platooning conditions. Furthermore, this can give an insight of the string stability of the system. Unfortunately, the platoon size in the experiments is too small to draw any definite conclusions about string stability. Tests with more than two SVEAs could give valuable information about this.

B. System Comparison

When comparing Figure 7 and 8, one large difference is the oscillations of the follower before the velocity step. Neither of the followers reached steady state before the step. However, the vision based platoon's follower is oscillating significantly more than the communication based one.

It is also clear that the amplitude of the oscillations is lower in experiment two after the step as well. This indicates that the stability of the communication based system is better. However, as mentioned in section IV-A, the follower's and the leader's oscillations are similar towards the end of the run time. This raises suspicion of the vision based system actually being more stable than the experiment indicates at first glance. To confirm this suspicion more experiments have to be conducted.

Another noticeable difference is the levels of overshoot, which are significantly smaller in experiment two, where the leader even has an undershoot. In a platoon, and in traffic control in general, a large overshoot is undesirable since it is amplified for the vehicles upstream the traffic flow. An undershoot, on the other hand, indicates a string stable system and is desirable. However, once again, the high overshoot of the follower in experiment one is a response to the overshoot of its leader, and could be smaller if the leaders step response were better.

When comparing the asymptotes of Figure 7 and 8, it is clear that the steady state errors are smaller in experiment two. The initial guess was that the reason for the large error in experiment one was hardware related and that the actual speed of the SVEAs might not directly correspond to the velocity in the system. However, the error in experiment two contradicts this guess since the same two SVEA vehicles were used for the two experiments. Another explanation for the difference has not been found.

C. Integration of Systems

One of the aims of the experiments was to get an indication about the possibility of an integration of the two systems in a fallback solution. Based on the previous discussion, the large overshoot and oscillations in the vision based system might be problematic. However more testing has to be done do get a more reliable result and decide the characteristics of the vision based system when reacting to a stable velocity step.

The scripts used for the two scenarios are very similar and in respect to that, the two systems are compatible. Set aside the stability, a fallback system can therefore be implemented. However, in order to integrate the systems, phantom leaders might have to be added like shown in Figure 5.

D. Future Work

With the experiments conducted and analyzed, some areas of improvement have been identified. One of these is to make sure both vehicles in the platoon reach a steady state before the velocity step. This would give more valid results in terms of stability. It would also be of interest to see the response of the vision based system when reacting to a more ideal step. This would allow for investigation of the true characteristics of the system.

Furthermore, it would be of interest to optimize the experimental setup to enable more consistent results between runs, a topic that was problematized in the beginning of this section.

Another possible extension of the project is to repeat the two experiments with more than two SVEAs. This could give indications of, or conclusions about, the string stability of the systems.

Finally, it would be interesting to integrate the two systems by going from a cooperative communication based platoon to a vision based one during a run and analyze this fallback system.

VI. Conclusion

Firstly, it can be stated that it is feasible to platoon using a camera and OVRV model but the system isn't as stable as desired. Secondly, it can be concluded that the cooperative communication based system is more stable than the vision based one. This would, however, have to be confirmed by further testing. Moreover, integration of the two systems might be problematic due to the difference in stability. Finally, there is potential for a fallback system but this has to be investigated by performing experiments on SVEAs with a program integrating the two systems.

Acknowledgment

The authors would like to thank their supervisor Frank Jiang for his endless dedication to make this bachelor thesis a educative, exiting and rewarding project. His patience and pedagogical approach inspires to learn.

References

[1] E.-E. Commission, *EU Transport in Figures — Statistical Pocketbook*. Brussels, Belgium: Publications Office of the European Union, 2020.

[2] P. E. Paré, E. Hashemi, R. Stern, H. Sandberg, and K. H. Johansson, "Networked model for cooperative adaptive cruise control," *IFAC-PapersOnLine*, vol. 52, no. 20, pp. 151–156, 2019, 8th IFAC Workshop on Distributed Estimation and Control in Networked Systems NECSYS 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896319320002

[3] L. Zhang, F. Chen, X. Ma, and X. Pan, "Fuel economy in truck platooning: A literature overview and directions for future research," *Journal of Advanced Transportation*, vol. 2020, pp. 1–10, Jan. 2020. [Online]. Available: https://doi.org/10.1155/2020/2604012

[4] Y. Wang, Z. Jiang, X. Gao, J.-N. Hwang, G. Xing, and H. Liu, "Rodnet: Radar object detection using cross-modal supervision," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2021, pp. 504–513.

[5] S. Benhimane, E. Malis, P. Rives, and J. Azinheira, "Vision-based control for car platooning using homography decomposition," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 2161–2166.
[6] S. Mitchell, I. Sajjad, A. Al-Hashimi, S. Dadras, R. M. Gerdes, and R. Sharma, "Visual distance estimation for pure pursuit based platooning with a monocular camera," in *2017 American Control Conference (ACC)*, 2017, pp. 2327–2332.
[7] O. Kalachev. (2021, Apr.) Aruco markers generator. [Online]. Available: https://chev.me/arucogen/
[8] G. Naus, R. Vugts, J. Ploeg, M. Molengraft, and M. Steinbuch, "String-stable cacc design and experimental validation: A frequency-domain approach," *IEEE Transactions on Vehicular Technology*, vol. 59, pp. 4268 – 4279, Dec. 2010.
[9] KTH-SML, "Svea starter," https://github.com/KTH-SML/svea_starter/tree/platooning_master, 2021.

Splitting and Merging of Platoons With the Help of PID Control

Marcus Didenbäck and Gustav Jonsson

***Abstract*—For an environmentally sustainable society the transportation of goods must be optimized. The next step in making truck transportation more sustainable is platooning. Platooning is when vehicles drive close together in a line which requires implementation of a control system. The aim of this project is to tune and implement a stabilizing controller that can handle various scenarios. This paper proposes a PID controller to solve the problem of achieving platooning. Using a simulation environment written in Python, the PID controller is tuned for three specific scenarios; platooning, merging and splitting. To obtain a smooth control signal setpoint weighting was implemented. The results presented in the report show that PID controller can create a stable platoon for the range of tested scenarios. However, these results are purely theoretical and while promising, more tests must be done to determine if the results hold in practice.**

***Sammanfattning:* För att kunna uppnå ett hållbart samhälle måste lastbilstransporter effektiviseras. Nästa steg i utvecklingen mot mer hållbara lastbilstransporter är konvojkörning. Konvojkörning syftar på att fordon kör nära varandra i led. Detta är dock inte möjligt för mänskliga chaufförer att åstadkomma och därmed krävs det något typ av kontrollsystem. Syftet med projektet är att konstruera en kontroller som kan hantera tre stycken scenarion, nämligen konvojkörning, separering och sammanslagning. Rapporten föreslår en PID-regulator för att hantera dessa scenarion. En simuleringsmiljö byggdes upp i Python och där justerades reglerparametrarna för att klara av uppgiften. För att göra PID-kontrollen stabil implementerades setpoint-weighting. Resultaten visar att PID-regulatorn kan erhålla en stabil konvoj för de utförda testerna. Resultaten är dock helt teoretiska och även om de är lovande måste fler tester göras innan konkreta slutsatser om hur lösningen fungerar i praktiken kan dras.**

***Index Terms*—Platooning, PID, control system, setpoint weighting, trucks.**

Supervisors: Xiao Chen, Miguel Aguiar

TRITA number: *TRITA-EECS-EX-2021:139*

I. Introduction

Freight transport has been a key pillar of our human civilization since the dawn of modern society. Over 70% of goods in the US are transported by trucks [1]. However, climate change requires us to increase the efficiency of truck transportation to make it sustainable and one of the next steps in the evolution of truck transport is platooning. Platooning trucks drive close to each other in long lines in order to reduce wind drag and thereby reduce fuel consumption, transportation cost, road accidents and also increase the comfort of truck drivers [2], [3]. In order to make platooning a reality the trucks will require control systems [4] that can precisely control trucks present on roads today [5]. One common control system today is the adaptive cruise control system (ACC) which has the purpose to keep a safe distance to the vehicles in front. Another control system is the cooperative adaptive cruise control system (CACC) which is an expansion of the ACC system. Instead of measuring data from other vehicles as ACC does, CACC uses vehicle-to-vehicle (V2V) communication to gather information from the surrounding vehicles, such as speed and location.

Platooning requires some variants of these control systems. The biggest difference between CACC and platooning according to [6] is how the system being controlled is constructed. A platoon is made up of a leader vehicle and a minimum of one follower vehicle that exchanges information between each other to maintain a safe and sustainable platoon in comparison with CACC where all vehicles are leader vehicles in their own ”platoon” [7].

To control a platoon a controller is required, for example a proportional-integral-derivative (PID) controller or a model predictive controller (MPC). In this project, the purpose is to develop a PID controller that can handle splitting and merging of the platoon.

II. Background

A. Literature

A number of vehicle platooning projects have previously been carried out. In [8] some of the projects are described, e.g SARTRE, PATH, GCDC, Energy ITS, and SCANIA-platooning. One of the main points in all of these projects was energy-saving combined with other specific goals, except GCDC, where the main point was to accelerate the progress and implementation of platooning using a combination of V2V and V2I (vehicle-to-infrastructure) communication together with top modern sensor fusion technology.

In SARTRE the main point was to construct a platoon with a leader vehicle in front that steers and controls the platoon. Its members are in turn autonomously controlled so they can perform other tasks while connected to the platoon. All this to get the benefits of platoon driving, e.g reduced fuel consumption and with the issue to handle other non-platoon vehicles. Energy ITS also had the aim of energy saving using platooning with the side goal to resolve the problems associated to the lack of skilled drivers.

PATH researched the ability to increase a traffic lane’s capacity using platooning. The studies showed that it could

be possible to increase the capacity of a lane by up to two or three times, if platoons with up to ten cars were implemented.

In the project SCANIA-platooning the aim was to reduce the fuel consumption using platoons to decrease the fuel costs for fleet owners as well as decreasing the damage done to the environment.

B. PID

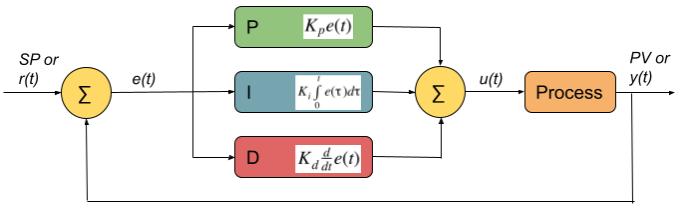


Fig. 1. A model of a PID controller.

A proportional-integral-derivative controller (PID controller) [9], [10], [11] consists of three parts: a proportional, an integral and a derivative part, which is why it's also refered to as a three-term controller. PID control is used for regulating a process using feedback from the output of the system. It works by calculating an error $e(t)$ between the reference value, usually called the setpoint value SP, also named as $r(t)$ and the actual value, usually called the process variable PV, also named $y(t)$. According to these values and the tuning parameters K_p, K_i and K_d an adjustment is applied to the output value $u(t)$. In Fig. 1 a block diagram of a PID controller can been seen with all these parameters mentioned above. The PID controller is written in equation (1)

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{d}{dt} e(t). \tag{1}$$

The time discrete version of (1) is written in equation (2)

$$u(k) = K_p e(k) + K_i \sum_{k'=0}^{k} e_{k'}(t_k - t_{k-1}) + K_d \frac{e_k - e_{k-1}}{t_k - t_{k-1}}. \tag{2}$$

The PID controller's three tuning parameters have different properties. All parameters have to be equal to or larger than zero, i.e they are non-negative parameters. Below all three parameters and their respective properties are described.

- Proportional constant, K_p: The proportional term only depends on the error value. Increasing the proportional constant results in a faster response of the process. If the constant becomes too large it may result in an unstable process.
- Integral constant, K_i: The integral term sums the error value over time and if there is some error part left after the proportional term has acted the integral term will try to minimize that error value. Without an integral part of the controller this error will not be minimized and the system will at its steady state not be at the desired value, this is called a static error.
- Derivative constant, K_d: The derivative term also called the prediction term uses the previous values to predict how the error will look like in the future. The derivative term is sensitive against noise and if the signal the derivative term acts on is too noisy the system may become unstable.

However, when the trucks split or merge their respective controller's setpoint values must change. This sudden change in setpoint value causes the output signal to suddenly jump because of the K_d term in equation (1). Since changes in the setpoint value are unavoidable for achieving split and merge the PID controller has been adjusted with setpoint weighting [12]. This leads to an output signal,

$$u(k) = K_p e_p(k) + K_i \sum_{k'=0}^{k} e_{k'}(t_k - t_{k-1}) + K_d \frac{e_d - e_{d-1}}{t_k - t_{k-1}} \tag{3}$$

where $e_p = \beta r - y$ and $e_d = \gamma r - y$. These two new parameters, β and γ, allows for more control in the tuning of the controller by controlling how much the setpoint influences the error signal that the derivative term and proportional term operates on. The integral term should be left to operate on the true error value in order to remove the static error completely. While equation (3) looks a lot like equation (2) the small difference allows the controller to handle the setpoint change smoothly.

Fig. 1 shows a block diagram of a regular PID controller and an important part of any control system is its transfer function. This is obtained by Laplace transformation of equation (1), which gives equation (4)

$$G(s) = K_p + \frac{K_i}{s} + sK_d. \tag{4}$$

The frequencies for which the denominator of the transfer function is zero are called poles and directly influence the system's stability and performance [13]. From equation (4) it can be observed that the integral part of the PID controller adds a pole to the system which can cause instability. This is why one should be careful when tuning the integral part of a PID controller, but it is also necessary in order to remove the static error of the system.

C. Platooning

In [14] and [15] platooning and its benefits are discussed. In platooning vehicles are connected to each other via communication systems. Through this communication network the vehicles can share their information to the other vehicles such as their speed, acceleration, length, weight and other useful information that could be of interest. Vehicles attached to the platoon should be able to leave and connect at any point.

Platooning reduces the fuel consumption and therefore also the fuel costs, as the air resistance reduces for all the follower trucks in the platoon. Moreover, the workload of the drivers in the follower trucks is reduced thanks to the autonomous systems. This leads to less time spent on monotone driving

for the drivers and should result in fewer accidents caused by fatigue.

III. Problem formulation

A. Terminology

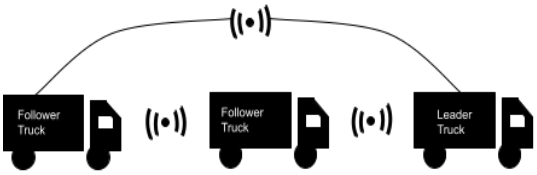


Fig. 2. Platoon of three trucks.

Fig. 2 depicts a platoon of three trucks to describe how the trucks will be named as in the rest of the report. The first truck in the platoon will be referred to as the leader truck while the rest will be referred to as follower trucks. For the purposes of this report the leader truck does not use a PID controller to control its speed, but rather it uses cruise control with disturbances we can control. Each follower truck communicates with the truck directly in front of it as well as with the leader truck.

B. Platoon maneuvers

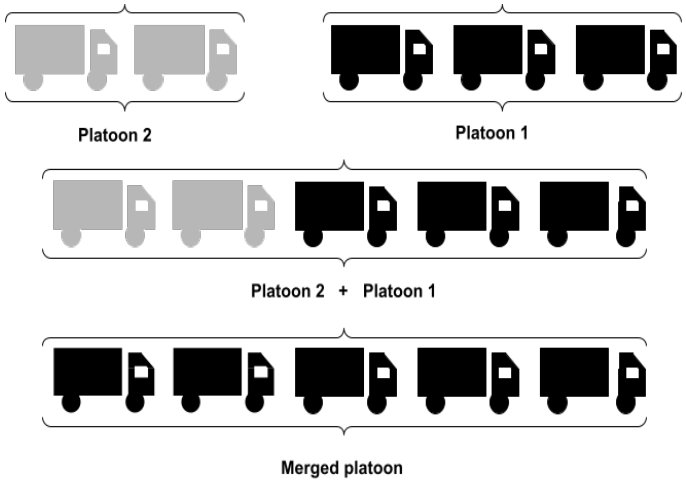


Fig. 3. Two platoons merging into one larger platoon.

1) Merging: Fig. 3 shows a merging scenario of two platoons with three and two trucks respectively, merging into one larger platoon. Merging is an important aspect in that a platoon must be able to handle since one single long platoon is more effective then several small ones, to gain the benefits of platooning. Merging is initialized when two platoons are close to each other. The leader truck of the second platoon will then become a follower truck of the first platoon and thereby close the distance as it approaches its new desired inter-vehicular distance. Thereafter the trucks will continue driving as one platoon.

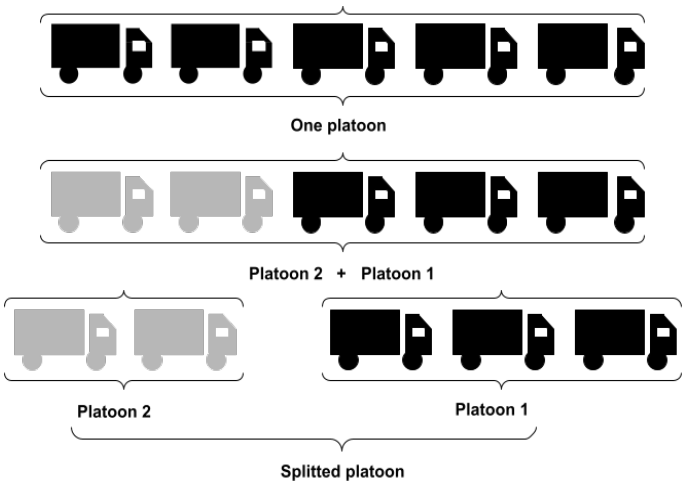


Fig. 4. One platoon splitting off into two smaller ones.

2) Splitting: When a platoon split is initiated one of the follower trucks will be assigned as the splitting truck. The splitting truck will increase its reference to the truck in front. Once complete, the splitting truck will become the leader truck for the new split-off platoon. This scenario is illustrated in Fig. 4.

IV. Modelling

As mentioned in III, each follower truck will communicate with the truck directly in front of them as well as the leader truck. This is to get information about the relative positions and the difference between the positions and the setpoint which in this report has been set to 5 meters between each truck. The error value can therefore be described as

$$e(t) = PV_L - SP_L + PV_N - SP_N, \tag{5}$$

where L and N index references leader truck and next truck respectively. This has been done because referencing two trucks in the platoon gives a more stable platoon overall whereas only referencing the distance to the truck in front will cause the error to propagate along the platoon for any disturbance.

For the follower trucks in the platoon the wind resistance is assumed to be negligible, therefore the force required to accelerate a follower truck can be described as,

$$ma = F_x - mg\sin\Theta - f_r mg\cos\Theta. \tag{6}$$

Where F_x is the traction force generated by the truck and Θ is the road grade. The rolling coefficient f_r is given empirically by [16].

The simulation environment creates a user-defined amount of trucks on a highway driving at 20 m/s $\approx$ 70 km/h in three different scenarios all of which have a time resolution of 20 ms.

1) A single platoon driving where the effects of disturbances and sudden changes in velocity can be observed and how the PID controller handles these.

2) Two platoons approaching each other and when they have reached a user-defined distance to one another the second platoon will merge into the first creating one single larger platoon.
3) One single platoon splitting a user-defined length somewhere between two trucks in the platoon and once the split has been achieved the trucks will continue as two separate platoons.

Situation three is especially useful for determining how well the controller can react to an outside vehicle suddenly entering the platoon or for merging a single truck that just entered the highway.

TABLE I
SIMULATION PARAMETERS.

Parameter	*Value*	*Parameter*	*Value*
K_p	1	β	1
K_i	0.2	γ	0
K_d	2	Θ	0

V. RESULTS

This section of the report will detail the results obtained from the simulation environment described in section IV. The first test evaluates the robustness of the controller with respect to disturbances in the leader truck's velocity. The obtained results are shown in V-A, it is followed by a simulation of the merging scenario in V-B and lastly a simulation of the splitting scenario is presented in V-C.

The following results have been obtained using parameters and their respective values given in Table I.

A. Platooning

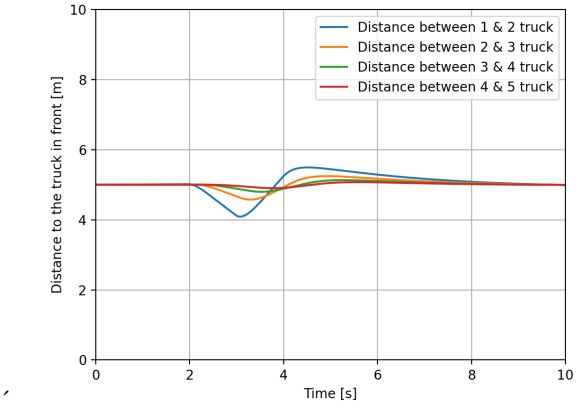


Fig. 5. Inter-vehicular distances when the leader truck's velocity was disturbed.

The disturbance induced on the leader truck was a sharp decrease in the leader truck's velocity followed by a sharp increase back to its original speed. As shown in Fig. 5 the platoon safely manages the disturbance, but one thing to note is that the spacing error between the trucks diminishes further back in the platoon and a longer platoon would thereby not cause any problems.

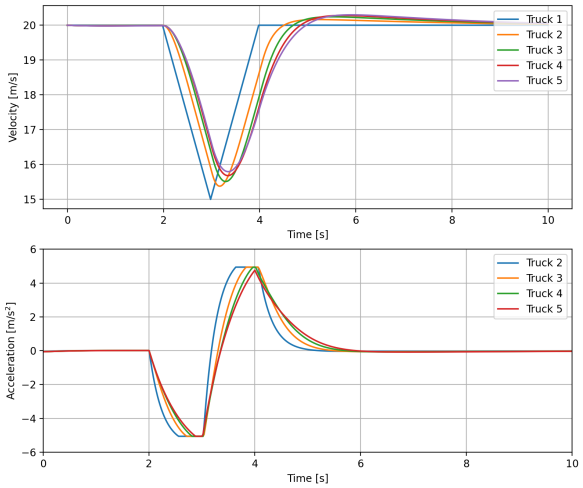


Fig. 6. Speed and acceleration of the trucks when the leader truck's velocity was disturbed.

From Fig. 6 the velocities and accelerations of the trucks during the test can be studied. Table I shows that the derivative part of the controller is the largest. This causes the system to be well dampened as can be observed from the plot with little overshoot and minimal oscillations which is crucial for a functioning platoon.

B. Merging

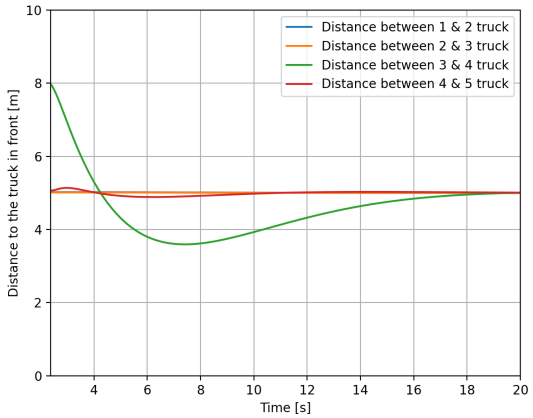


Fig. 7. Inter-vehicular distances between the trucks in the merged platoon.

Fig. 7 and 8 shows the performance of the PID controller for the merge maneuver of two platoons. The first platoon consists of three trucks and the second platoon consists of two trucks. In Fig. 7 the inter-vehicular distances between the trucks are shown when the merge is activated. The setpoint

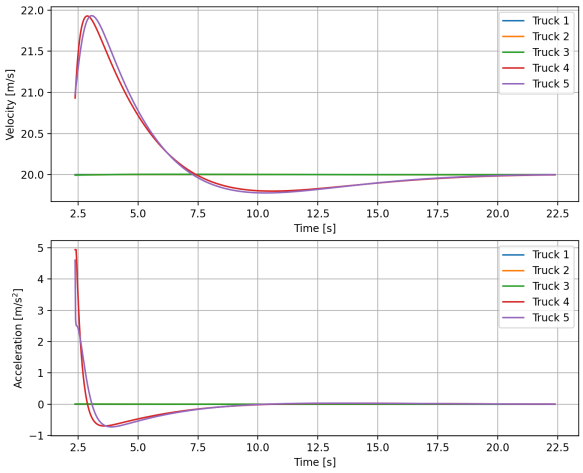


Fig. 8. Speed and acceleration of the trucks during the merge maneuver.

value for the merge maneuver to activate is set to 8 meters. The PID controller handles this maneuver very well, as can be seen in the small oscillations in the distances between the followers. The highest merging distance before the merging platoon crashes into the first platoon was found to be 17 meters.

In Fig. 8 the velocity and acceleration timelapse during the merge maneuver is stated. The trucks in the merged platoon have a short delay under a second where the velocity of the trucks are increasing, before its decreasing smoothly to the velocity of the first platoon.

C. Splitting

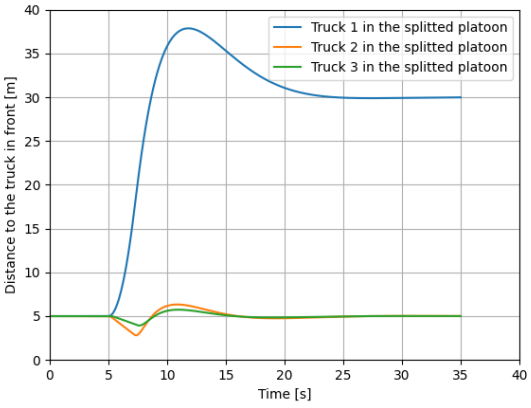


Fig. 9. Inter-vehicular distances between the trucks during the split maneuver.

Fig. 9 shows a platoon splitting with a split length of 25 meters as well as the reaction of the two trucks after the splitting one. From the splitting truck's distance to the truck in front, an overshoot of around eight meters can be seen. From a safety perspective, a larger inter-vehicular distance is not a problem, but Fig. 9 shows that the truck directly behind the splitting truck keeps the shortest distance to the truck in front and will be the critical part of whether or not the split will be successful. In other words, if truck 2 does not crash into truck 1 no truck will crash and if any truck is going to crash truck 2 will crash first. For a 25 meter split distance truck 2 still manages to keep a distance of above 3 meters to truck 1.

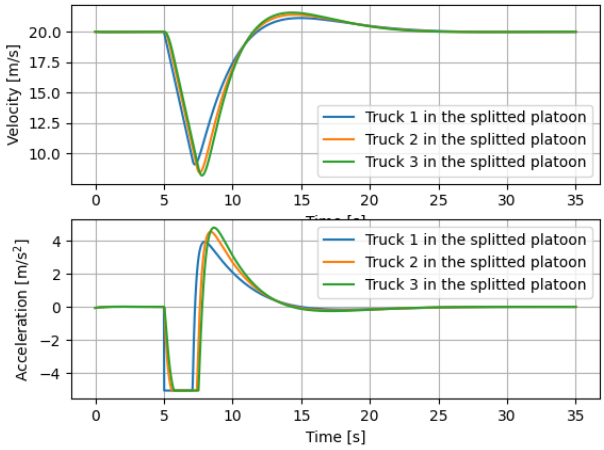


Fig. 10. Speed and acceleration for the trucks during the split maneuver.

Fig. 10 shows the velocity and acceleration of the trucks during the split maneuver with a 25 meter split distance. From this, we can see the effects of setpoint weighting described in II-B. When the splitting commences the setpoints for the trucks in the splitting platoons must change, but this does not lead to any discontinuities in the acceleration.

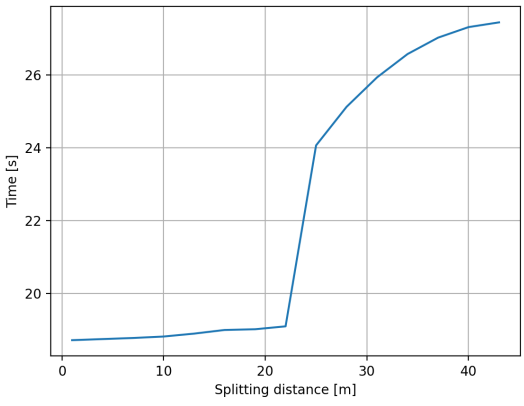


Fig. 11. Time to achieve the split for the allowed splitting distances.

Fig. 11 shows the time it takes to achieve different splitting distances. The splitting is defined as complete when the error in the spacing error is less then 2%. The sudden increase in splitting time at approximately 22 meters split distance occurs

because that is the point at which the spacing error oscillates one time before settling.

VI. Discussion

To achieve platooning many different control systems could be used. In this project, a PID controller was designed and implemented to study the controller's behavior for the maneuvers described in III. The discussion is carried out from the results obtained in V. In section VI-A the advantages of the PID controller and why a PID controller was chosen are discussed. Besides the advantages, the limitations of implementing and tuning a PID controller are discussed in section VI-B. In section VI-C, a sustainability point of view of the PID controller has been reflected. Finally, in section VI-D, considerations of interesting aspects to add to future work are mentioned.

A. Advantages

The reason for choosing a PID controller is because it is the most common [11] controller used in the industry due to its cheap and simple design and ease of implementation. This is where the PID controller shines, in its simplicity. One of the advantages of using a simple control system is that it can easily be changed/tuned to fit specific scenarios. A truck driver could for instance be able to follow instructions such as, in case of rainy weather, turn dial 3 to the second notch, without any specific training. This dial could then represent one of the tuning variables of the PID, whereas if a more advanced control system was implemented, changing and tuning it would most likely require a technician.

From the results in section V, especially in Fig. 5, 7 and 9, it's easy to discern that it's the distance when the merge is activated and the splitting distance that is the critical part for the PID controller to handle and not the number of trucks in each respective platoon. The critical distance for the merge maneuver is 17 meters and 44 meters for the splitting maneuver, which is good enough. There is no need to start the merge earlier and the splitting distance doesn't need to be any longer either, because the allowed distance is almost double that of the allowed maximum length of a truck on public roads today [5].

B. Limitations

The choice of the PID controller limited the ability to define constraints for the problem. The only constraints that could be set were the maximum and minimum acceleration which corresponds to full throttle and hard braking respectively, to keep the simulation realistic. This is in contrast to for example a model predictive controller where certain constraints can be incorporated directly into the control design. Whereas instead, a PID controller needs to be tuned to make sure it does not break any constraints one might wish to have. The method that has been used in this project to see that the controller fulfills the requirements is the trial and error method. It's a method where the controller's parameters are tuned one at a time until the results are satisfactory.

Since the controller can't set constraints it doesn't have any real safeguard against collisions between the vehicles in the platoon. Therefore the controller must be tested thoroughly for many different cases to see that it can handle various cases. This could be a big problem when the platoon is performing on real roads, where there is no room for failure.

Another downside of the controller is that it does not take time into consideration. This results in large retardation of the truck even for small splitting distances or large accelerations for small merges. Other than possibly being quite unpleasant for the driver of the truck this is not an eco-friendly driving technique.

C. Sustainability

One of the most interesting aspects in developing controllers for platooning on public roads is the environmental point of view. Trucks driving in platoons will drastically decrease fuel consumption due to the reduced wind drag and therefore also decrease the carbon dioxide emissions. Due to reduced fuel consumption, transportation with trucks in platoons will lead to reduced fuel costs. This could potentially lead to an increase in the use of truck transportation thanks to its economic advantages. This could lead to a transition from other transportation methods which are more sustainable than truck transportation and maybe not result in a net positive for the environment.

Achieving platooning will also lead to less congested roads and fewer traffic jams thanks to the very compact driving of the trucks. Reducing traffic congestion can greatly help in reducing carbon dioxide emissions as well as air pollution [3]. However, while less trafficked roads are good for the environment this could possibly backfire by more people choosing their car as transport instead of public transport or bikes since the roads are not as congested anymore.

D. Future work

For future work there are some aspects that could be interesting to look into. One of them is to expand this model which is only based on the longitudinal direction to also manage these maneuvers in the horizontal direction to get an even more realistic look at the problem. It could also be interesting to try other methods to find and tune the parameters of the controller, instead of the trial and error method that is used in this project, to see how much it would be possible to improve the controller to keep it stable for longer distances between the platoons and to handle more unprepared disturbances.

Another aspect that could be interesting to investigate is the efficiencies versus the cost for different controllers to see how they relate to each other.

VII. Conclusion

From the results above we can draw the following conclusions:

- Despite its simplicity, a PID controller can control a platoon of trucks on the highway quite efficiently.
- The lack of strict safeguards against collisions can make implementing a pure PID controller difficult as long as humans are still on the roads.
- The PID controller must be well tested for all scenarios due to the difficulty in setting constraints, for this a robust simulation environment must be created.
- The controller does not take time into consideration and can cause sporadic changes in the acceleration when not completely necessary.
- Under ideal conditions, a PID controller is sufficient to control a platoon of trucks on a highway.

Acknowledgment

The authors would like to thank their supervisors Jonas Mårtensson, Xiao Chen and Miguel Aguiar.

References

[1] S. John. (2019, Jun) 11 incredible facts about the $700 billion us trucking industry. [Online]. Available: https://markets.businessinsider.com/news/stocks/trucking-industry-facts-us-truckers-2019-5-1028248577

[2] F. Coniglio, *A holistic view on the consequences of Truck Platooning*. Saarbrücken: AV Akademikerverlag, Oct 2017.

[3] P. Hao, C. Wang, G. Wu, K. Boriboonsomsin, and M. Barth, "Evaluating the environmental impact of traffic congestion based on sparse mobile crowd-sourced data," in *2017 IEEE Conference on Technologies for Sustainability (SusTech)*, 2017, pp. 1–6.

[4] Z. Wang, G. Wu, and M. J. Barth, "A review on cooperative adaptive cruise control (cacc) systems: Architectures, controls, and applications," pp. 2884–2891, 2018.

[5] S. Larsson. (2009, Jun) Weight and dimensions of heavy commercial vehicles as established by directive 96/53/ec and the european modular system (ems). [Online]. Available: https://ec.europa.eu/transport/sites/transport/files/modes/road/events/doc/2009_06_24/2009_gigaliners_workshop_acea.pdf

[6] X. Lu and S. Shladover, "Integrated acc and cacc development for heavy-duty truck partial automation," in *2017 American Control Conference (ACC)*, 2017, pp. 4938–4945.

[7] T. S. T. Administration. (2020, May) Platooning med lastbilar. [Online]. Available: https://www.trafikverket.se/om-oss/nyheter/aktuellt-for-dig-i-branschen3/aktuellt-om-forskning-och-innovation2/2020-05/konvojkorningplatooning-med-lastbilar/

[8] Bergenheim, C.; Shladover, S.; Coelingh, E. (2012, Oct.) Overview of platooning systems. Vienna, Austria. Proceedings of the 19th ITS World Congress. [Online]. Available: http://publications.lib.chalmers.se/publication/174621

[9] Kiam Heong Ang, G. Chong, and Yun Li, "Pid control system analysis, design, and technology," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 559–576, 2005.

[10] N. I. Corporation. (2020, Mar) Pid theory explained. [Online]. Available: https://www.ni.com/sv-se/innovations/white-papers/06/pid-theory-explained.html

[11] E. P. Focus. (2021, Apr) What is a pid controller : Working its applicationsd. [Online]. Available: https://www.elprocus.com/the-working-of-a-pid-controller/

[12] H. T. Åström, Karl J., "3.4.2 setpoint weighting," 1995. [Online]. Available: https://app.knovel.com/hotlink/khtml/id:kt007XPDSE/pid-controllers-theory/setpoint-weighting

[13] S. Torkel Glad, Lennart Ljung, *Reglerteknik Grundläggande teori*. Poland: Studentlitteratur AB, 2018.

[14] Dellrud, J. (2020, Jan.) Sweden4Platooning. . . [Online]. Available: https://www.trafikverket.se/contentassets/2885e60e4b7448e393b4cd5a66b461b3/publik-rapport_tsaf_20200130.pdf

[15] European Automobile Manufacturers Association. (2017) What is truck platooning? [Online]. Available: https://www.acea.be/uploads/publications/Platooning_roadmap.pdf

[16] J. Y. Wong, *Theory of ground vehicles, 3rd ed.* New York: John Wiley Sons, 2001.

Splitting a Platoon Using Model Predictive Control

Albin Gustafsson and Emil Vardar

***Abstract*—When multiple autonomous vehicles drive closely together behind each other, it is called a platoon. Platooning provides several benefits, such as decreased congestion and reduced fuel consumption. In order for more vehicles to take advantage of these benefits, the platoon should be able to open up a space for other vehicles to merge into. Thus, our goal with the project was to develop a system that can split a platoon. To achieve this, we are using model predictive control (MPC) to control the system because it can handle constraints and control systems with multiple variables. To test the implemented system, we created a simulation environment in Python. We created several plots to analyze and show the results of the simulations. To make the simulation more realistic, we introduced air drag to the system. To counteract this effect, we added linearized air drag to the MPC. We showed that the constructed system could split between any two adjacent vehicles in a platoon up to 50 meters. Another significant result was that the MPC could compensate for the air drag without adding linearized air drag to the MPC.**

***Sammanfattning*—När flera autonoma fordon kör nära varandra kallas det för en platoon. Det finns flera fördelar med platooning som minskad trafik samt minskad bränsleförbrukning. För att fler fordon ska kunna dra nytta av dessa fördelar bör nya fordon kunna sammansluta till en platoon och på grund av detta bör fordonen i platoonen kunna öppna ett utrymme för det nya fordonet. Därför är vårt mål med detta projekt att utveckla ett system som kan styra och dela på en platoon. För att åstadkomma detta använder vi model prediktiv reglering (MPC) eftersom den är bra på att hanterar bivilkor och styra system med många variabler. Vi implementerade systemet i Python, där en simuleringsmiljö skapades. För att se och analysera resultaten av simuleringen skapades grafer som visade hur fordonen hade färdats under simuleringen. Vi lade till luftmotstånd i simuleringen för att göra den mer realistisk. För att motverka luftmotståndet lade vi även till ett linjäriserat luftmotstånd till i MPC:n. I slutet av projektet kunde systemet dela platoonen mellan två fordon med ett avstånd upp till 50 meter. Vi observerade att MPC:n kunde kompensera för luftmotståndet utan implementationen av det linjäriserade luftmotståndet.**

***Index Terms*—Model Predictive Control (MPC), Platooning, Splitting, (Linearized) air drag, Constraints.**

Supervisors: Xiao Chen and Miguel Aguiar

TRITA number: TRITA-EECS-EX-2021:140

I. Introduction

Today many semi-autonomous vehicles are already in use on roads around the globe. Research regarding autonomous vehicles has advanced rapidly in the last couple of years. Autonomous vehicles provide several benefits such as decreased congestion and, perhaps most importantly, increased safety for all road users. When multiple autonomous vehicles drive closely together behind each other through communication, it is called a platoon.

This report is divided into seven sections, starting with section I, where an introduction and some background are given. Then the problem is formulated and our assumptions are given in section II. In section III the theory used in this project is explained both for the system's dynamics and the MPC. Section IV builds upon the previous section to describe how the theory is used to create the desired control system. In section V the results that we obtain from the simulations are presented and discussed. After that, in section VI, we propose some further improvements to the system. Finally, in section VII a conclusion to the project is given.

A. Background

In a platoon, all vehicles experience decreased air drag, resulting in improved fuel efficiency [1]. In [2], Valerio Turri estimates that up to a third of a truck transport company's total revenue is spent on fuel. If truck platooning reduces fuel consumption by a few percent, it will save a significant amount of money for the company. Many trucks travel on the same long stretches of roads, making platoons useful because of the economic benefits associated with the platoons. Apart from the economic benefits, it also reduces greenhouse gas emissions which is essential for meeting the environmental milestones.

Splitting a platoon could be useful for several reasons. For example, assume a vehicle is coming from a ramp and wants to enter the highway, but a long platoon is blocking the ramp, such as in figure 1. In this scenario, the platoon should be able to split so that the vehicle on the ramp can enter the highway without stopping or colliding with the platoon. Another example is to let a new vehicle merge into the platoon, so the merging vehicle can also take advantage of the platoon's benefits.

In [2] and [3] the authors showed that a platoon of vehicles has a lower air drag coefficient than if they were driven separately. This is because of the short inter-vehicular distances. In [1], Assad Alam underlines that even the leading vehicle's air drag coefficient decreases in a platoon. The dynamics for a platoon are quite complex and contain many variables that need to be optimized to reach the desired outcome. One of the early writings in platooning is [4] which proposes a centralized controller. The same logic is still used today, e.g. in [5]. In this paper, Duret et al. propose a higher level tactical layer that decides the order of vehicles in a platoon and between which vehicles a split should be created. The tactical layer then sends these instructions to a model predictive controller (MPC), responsible for calculating the optimal control signals for the vehicles to achieve a split. Furthermore, in [5] non-autonomous vehicles' merging scenarios are also taken into account, which is out of the scope of our project. Atsushi et al. have created a program for controlling a vehicle around a track using MPC [6]. This code and the code in [7], where MPC has been implemented in Python, have been a great help

in this project. Merging and splitting have also been tested by using a PID controller in [8]. In [9], Assad Alam et al. use a decentralized controller for managing a platoon with an LQR controller.

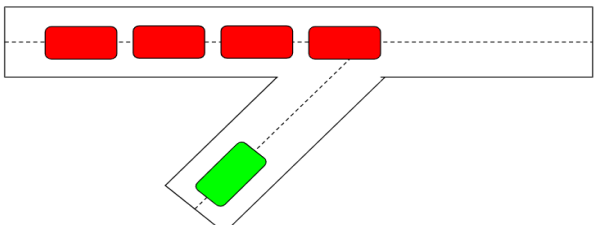


Fig. 1. A platoon blocking a highway ramp when a vehicle coming from a ramp, wants to enter the highway.

B. Choice of controller

Model predictive control (MPC) has shown to be efficient for platooning. This is because an MPC can predict the future by using a state-space model (see section III) and accordingly obtain the optimal solution for the present time. On top of this, MPC can control a system with multiple inputs and multiple control signals [10]. This is important for controlling a platoon since the controller gets data from all the vehicles in the platoon resulting in multiple inputs. Moreover, it must control all the vehicles resulting in multiple control signals. While the MPC is calculating the optimal solution, it also takes into account the given constraints [10]. This is very important in a platoon where many constraints are present. Based on all of the mentioned features for MPC above, we chose it as our controller in this work.

II. Problem formulation

A. Properties that a control system should have

1) A control system should be able to split between any two vehicles: As mentioned in the previous section, a platoon should be able to split for many reasons. In most of these scenarios, a split can be required anytime and between any two vehicles to prevent accidents or enhance the driving experience. For example, depending on how fast the green vehicle is traveling in figure 1, the split should be able to initiate between different vehicles.

2) A control system should be able to split up to a certain distance: The merging vehicles can have different lengths. That is why the split should be able to manage various split distances. In Sweden, the maximum permitted length for a vehicle that can be driven on public roads is $25.25\,\mathrm{m}$ [11]. For a vehicle with length $25.25\,\mathrm{m}$ to merge into a platoon, a gap of approximately $50\,\mathrm{m}$ is needed ($10\,\mathrm{m}$ margin to the vehicle in front and $10\,\mathrm{m}$ of margin to the vehicle behind). Therefore, we set the upper limit to $50\,\mathrm{m}$ in this project.

3) A control system should be adjustable according to the number of vehicles present in a platoon: There is no upper limit for how many vehicles can participate in a platoon. In fact, the more vehicles there are in the platoon, the better. Therefore, a system should be able to control a platoon with an arbitrary number of vehicles.

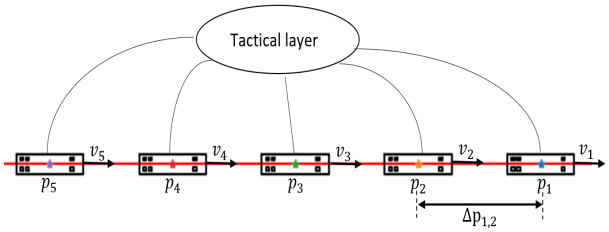


Fig. 2. A platoon of five vehicles which communicate with the tactical layer. Here v_i is the velocity and p_i is the position for the i:th vehicle.

4) A control system should satisfy necessary constraints: An appropriate control system that manages the split must take into account the constraints of the physical world. For example, an engine or a motor can only generate limited torque. Furthermore, a control system used for platooning should not overrule any of the legal speed limits [12]. No measuring instruments can measure a value with 100% accuracy. Taking this into account, a safety distance that can not be exceeded should be added to the system [12]. If the measurements are wrong, this safety distance might be exceeded in reality, but hopefully, the rear-end collisions will be avoided thanks to this safety distance.

5) A control system should consider the disturbances acting on the platoon: There are many disturbances present in the real world which the system should consider. The dominant one among them is the air drag. Air drag accounts for almost 80% of all resistances present on a medium-size car traveling at $100\,\mathrm{km/h}$ [3], [13]. The percentage will be less for the following vehicles in a platoon, but the air drag is still the biggest among all the losses. Therefore, our system must consider the air drag since we assume that the vehicles are moving with a velocity around $90\,\mathrm{km/h}$.

B. Tactical layer

In this project, we take a similar approach to [5]. That is, we are also using a centralized tactical layer, as can be seen in figure 2. An advanced tactical layer should analyze the road and the vehicles that are present in a system. Then, as a result of these, it should decide between which vehicles the split should be initiated and when the split should start and be done. Nevertheless, in this project, this information is manually given by a user. With the information obtained from the user, the tactical layer calculates the required values for the MPC. Then the required information is given to a centralized MPC (which can be seen as a part of the tactical layer). Having a centralized MPC makes the project more manageable than having a distributed MPC (one MPC in each vehicle).

Note that all the vehicles in the platoon inform the tactical layer about their absolute positions p_1, p_2, ..., p_n and velocities v_1, v_2, ..., v_n. Then this tactical layer obtains the optimal solution (using the MPC) for performing the split and returns the control values to each vehicle. In this project, the control signals are the accelerations for each vehicle (except the leading vehicle). The tactical layer is responsible for all the calculations and is implemented in Python. The final code is given in [14].

C. Assumptions

In this project, we assume that all vehicles have the same mass m, frontal area A, length L, air drag coefficient $C_{d,0}$ and engine/motor (thus the highest acceleration that can be given is the same). Furthermore, the lead vehicle is not controlled by the centralized controller. We assume that the leader travels with a constant velocity even if we add air drag to the system. The MPC controls all the other vehicles in the platoon. We also assume that values on position and velocity are measured correctly from the sensors and radars. The vehicles travel on a straight road, i.e., the vehicles can not turn or change lanes. As mentioned above, air drag accounts for up to 80% of all losses acting on a vehicle traveling at $100\,\mathrm{km/h}$. Therefore in this project, we assume that the air drag is the only outside force acting on the vehicles, i.e., gravitational force when driving up/down a hill and rolling resistance are assumed to be $0\,\mathrm{N}$.

III. Theory

A. Dynamics of the platoon

In a platoon, each vehicle has a different position, velocity, and acceleration. In this paper, we denote them as p_i, v_i and a_i, where i indicates the number of the vehicle. We assume the dynamics of vehicle i to be

$$\dot{p}_i(t) = v_i(t), \tag{1a}$$
$$\dot{v}_i(t) = a_i(t). \tag{1b}$$

These dynamics are essential when the state vector $\vec{x}(t)$ and the state-space representation are created.

The main purpose of this project is to bring the distances between the vehicles, to desired values. Therefore, a reference vector containing the target values must be created. The purpose of this reference vector is, to make the entries in $\vec{x}(t)$ approach the entries in the reference vector. Thus, the state vector $\vec{x}(t)$ and the reference vector $\vec{x}_{ref}(t)$ are as following

$$\vec{x}(t) = \begin{bmatrix} \Delta p_{1,2} \\ \Delta p_{2,3} \\ \vdots \\ \Delta p_{n-1,n} \\ v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}, \quad \vec{x}_{ref}(t) = \begin{bmatrix} \Delta p_{ref,1,2} + L \\ \Delta p_{ref,2,3} + L \\ \vdots \\ \Delta p_{ref,n-1,n} + L \\ v_1 \\ v_1 \\ \vdots \\ v_1 \end{bmatrix}, \tag{2}$$

where $\Delta p_{i,i+1} = p_i - p_{i+1}$, i.e. the distance between the center of vehicle i and $i+1$, and v_1 is the constant velocity of the leading vehicle and is set to $v_1 = 90\,\mathrm{km/h}$. $\Delta p_{ref,i,i+1}$ is the reference distance between vehicle i and $i+1$. Observe here that the goal is to get the distance between the rear end of the preceding vehicle and the front end of the following vehicle (inter-vehicular distance) to be $\Delta p_{ref,i,i+1}$. Therefore, it is important to take the length of the vehicles into account, this is why the constant L has been added to $\Delta p_{ref,i,i+1}$ in equation (2). In this work the length of the vehicles are $L = 12\,\mathrm{m}$.

The leading vehicle's velocity, v_1, has been used as reference since the goal is to move the vehicles in the platoon with the same speed to keep the inter-vehicular distances constant. This is under the assumption that the vehicles do not want to split. When a split is performed, the $\Delta p_{ref,i,i+1}$ is going to change, and to meet this value, the velocities for some vehicles need to diverge from v_1. This is not a big issue since the entries in $\vec{x}_{ref}$ are classified as soft constraints and can be overruled. In a platoon with n vehicles $\vec{x}(t)$ and $\vec{x}_{ref}(t)$ are vectors of length $2n-1$.

To manage the splits, the accelerations of the vehicles are being controlled. To get the accelerations into the picture, we include the velocities v_i in the state vector. This is because, when the state-space representation is obtained, the $\vec{x}(t)$ is derived. From equation (1b) we know that the derivative of v_i is a_i. The controlled accelerations, except the leading vehicle's, are being gathered in a control vector, here denoted with $\vec{u}(t)$. The control vector does not contain the leading vehicle's acceleration, because the leading vehicle is not controlled by the system. Thus, the control vector is given by

$$\vec{u}(t) = \begin{bmatrix} u_2, u_3, \ldots, u_n \end{bmatrix}^T. \tag{3}$$

Observe here that u_i represents the acceleration that is given to vehicle i. In other words, it is not the total acceleration acting on vehicle i. The total acceleration acting on the vehicle is going to be u_i minus the air drag. This has been explained in depth in section III-B. For a platoon with n vehicles $\vec{u}(t)$ is a vector of length $n-1$.

Lets us now derive the state vector, $\vec{x}(t)$, to build up the state-space representation for the system. The derivative of $\vec{x}(t)$ becomes

$$\dot{\vec{x}}(t) = \begin{bmatrix} v_1 - v_2 \\ v_2 - v_3 \\ \vdots \\ v_{n-1} - v_n \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}. \tag{4}$$

Now we separate this equation so that $\vec{x}(t)$ and $\vec{u}(t)$ end up separately. Here we assume that the system is under ideal conditions. Thus, no air drag is acting on the system. In this case $u_i = a_i$. Then the state-space representation can be written as

$$\dot{\vec{x}}(t) = \mathbf{A}\vec{x}(t) + \mathbf{B}\vec{u}(t), \tag{5}$$

where $\mathbf{A}$ is

$$\mathbf{A}^{(2n-1)\times(2n-1)} = \begin{bmatrix} \mathbf{0}^{(n-1)\times(n-1)} & \mathbf{K}^{(n-1)\times n} \\ \mathbf{0}^{n\times(n-1)} & \mathbf{0}^{n\times n} \end{bmatrix}. \tag{6}$$

The notation $\mathbf{A}^{(2n-1)\times(2n-1)}$ denotes that the matrix $\mathbf{A}$ has the dimensions $(2n-1)\times(2n-1)$. This notation will be used throughout this paper. Furthermore, $\mathbf{0}$ denotes a zero matrix. $\mathbf{K}$ given in equation (6) can be expressed as following

$$\mathbf{K} = \begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \end{bmatrix}. \tag{7}$$

The $\mathbf{B}$ matrix is given by

$$\mathbf{B}^{(2n-1)\times(n-1)} = \begin{bmatrix} \mathbf{0}^{n\times(n-1)} \\ \mathbf{I}^{(n-1)\times(n-1)} \end{bmatrix}, \tag{8}$$

where $\mathbf{I}$ is the identity matrix. $\mathbf{A}$ and $\mathbf{B}$ given in equation (5) has to be discretized since the MPC is implemented in discrete time. This is done by using the forward Euler method as follows

$$\mathbf{A}_d = \mathbf{I} + \Delta t\mathbf{A} \quad \text{and} \quad \mathbf{B}_d = \Delta t\mathbf{B} \tag{9}$$

where $\mathbf{I}$ is the identity matrix with the same dimensions as $\mathbf{A}$, thus also as $\mathbf{A}_d$. Equation (5) can then be represented as

$$\vec{x}(k+1) = \mathbf{A}_d\vec{x}(k) + \mathbf{B}_d\vec{u}(k), \tag{10}$$

in discrete time. Here k is the time instant. Observe here that the time derivative has changed to $k+1$ according to the forward Euler method.

B. Adding air drag to the system

When air drag is introduced to the system the total acceleration acting on vehicle i becomes

$$a_i = u_i - a_{i,drag}, \tag{11}$$

where $a_{i,drag}$ is the air drag acceleration acting on vehicle i. Note that, in this report when air drag is mentioned we are referring to the air drag acceleration, and not to the air drag force. Thus the dimension for the air drag in this paper is $\mathrm{m/s^2}$. The air drag acceleration depends on the inter-vehicular distances and the velocities, and is given by

$$a_{drag,i}(\Delta p_{i-1,i}, v_i) = \frac{\rho A}{2m} v_i^2 C_d(\Delta p_{i-1,i}). \tag{12}$$

Here ρ is the air density, A is the cross-sectional area of the vehicle and m is the vehicle's mass. $C_d(\Delta p_{i-1,i})$ is the air drag coefficient and is given by

$$C_d(\Delta p_{i-1,i}) = C_{d,0}\left(1 - \frac{k_a}{k_b + \Delta p_{i-1,i}}\right), \tag{13}$$

where $C_{d,0}$, k_a and k_b are constants [2]. From this equation, we see that when the inter-vehicular distance decreases, C_d also decreases.

C. Linearizing the air drag

The MPC should consider the air drag acting on the system to be able to counteract its effect. In order to make the MPC consider the air drag, the air drag needs to be added to the MPC via the state-space representation. We decided to add the $linearized$ air drag to the MPC. This is because adding the real air drag into the MPC would make the MPC nonlinear, and thus, it would be too complicated. Since the air drag depends on $\Delta p_{i-1,i}$ and v_i, a multi-variable Taylor-polynomial was used to linearize the air drag, given in equation (12). After multi-variable derivation and simplification, a linearized approximation of the air drag was obtained, such as

$$\tilde{a}_{drag,i}(\Delta p_{i-1,i}, v_i) = v_i T_i + \Delta p_{i-1,i} S_i + U_i. \tag{14}$$

The expressions for T_i, S_i and U_i are given in Appendix A. These values are constantly changing when the MPC is predicting the future, since they depend on previous values on $p_{i-1,i}$ and v_i at each time step (see section III-D). When linearized air drag is added to the MPC, the derivative of the state vector will be as in equation (4). However, now the $a_i \neq u_i$, instead it will be as in equation (11). Furthermore, $a_{i,drag}$ in this equation is approximated with equation (14). This results in

$$\dot{\vec{x}}(t) = \begin{bmatrix} v_1 - v_2 \\ v_2 - v_3 \\ \vdots \\ v_{n-1} - v_n \\ 0 \\ u_2 - v_2T_2 - \Delta p_{1,2}S_2 - U_2 \\ u_3 - v_3T_3 - \Delta p_{2,3}S_3 - U_3 \\ \vdots \\ u_n - v_nT_n - \Delta p_{n-1,n}S_n - U_n \end{bmatrix}. \tag{15}$$

Now when this is separated similar to equation (5), the $\mathbf{A}$ given in equation (6) extends to

$$\mathbf{A}^{(2n-1)\times(2n-1)} = \begin{bmatrix} \mathbf{0}^{(n-1)\times(n-1)} & \mathbf{K}^{(n-1)\times n} \\ \mathbf{0}^{1\times(n-1)} & \mathbf{0}^{1\times n} \\ \mathbf{S}^{(n-1)\times(n-1)} & \mathbf{T}^{(n-1)\times n} \end{bmatrix}, \tag{16}$$

where

$$\mathbf{S} = \begin{bmatrix} -S_2 & & \\ & \ddots & \\ & & -S_n \end{bmatrix}, \mathbf{T} = \begin{bmatrix} 0 & -T_2 & & \\ \vdots & & \ddots & \\ 0 & & & -T_n \end{bmatrix}, \tag{17}$$

and $\mathbf{K}$ is the same as in equation (7). In equation (15) a constant term U_i appears. Since this constant term is not multiplied with $\Delta p_{i-i,i}$ or v_i, it has to be added as a separate matrix such as

$$\mathbf{D}^{(2n-1)\times 1} = \begin{bmatrix} \mathbf{0}^{n\times 1} & -U_2 & \dots & -U_n \end{bmatrix}^T. \tag{18}$$

Thus, the state-space representation becomes

$$\dot{\vec{x}}(t) = \mathbf{A}\vec{x}(t) + \mathbf{B}\vec{u}(t) + \mathbf{D}. \tag{19}$$

The new state-space model is then discretized in the same way as before. However, this time the $\mathbf{D}$ matrix also needs to be discretized. The extended discretization is given by

$$\mathbf{A}_d = \mathbf{I} + \Delta t\mathbf{A}, \quad \mathbf{B}_d = \Delta t\mathbf{B} \quad \text{and} \quad \mathbf{D}_d = \Delta t\mathbf{D}. \tag{20}$$

D. Overview of Model Predictive Controller

As mentioned in section I-B, Model Predictive Control (MPC) is a control algorithm that optimizes the inputs to a system so that the desired outputs can be obtained. It predicts future inputs and outputs of a system, and optimizes at regular time intervals Δt. In this project, the MPC solves an optimization problem by minimizing a quadratic cost-function subject to a set of constraints. The cost-function $J(\vec{x} - \vec{x}_{ref}, \vec{u})$

is minimized with respect to $\vec{u}$ as

$$\min_{\vec{u}} \quad J(\vec{x}-\vec{x}_{ref},\vec{u}) \tag{21a}$$

$$\text{s.t.} \quad \vec{x}(k+1)=\mathbf{A}_d\vec{x}(k)+\mathbf{B}_d\vec{u}(k)+\mathbf{D}_d, \tag{21b}$$

$$1+L\leq \Delta p_{i,i+1}(k)\ [\text{m}],\ i=1,\dots,n-1 \tag{21c}$$

$$40\leq v_i(k)\leq 120\ [\text{km/h}],\ i=1,\dots,n \tag{21d}$$

$$-10\leq u_i(k)\leq 5\ [\text{m/s}^2],\ i=2,\dots,n \tag{21e}$$

$$\text{if } k_s<k+N \text{ and } k<k_s-3:$$

$$\Delta p_{ref,s,s+1}+L+1\leq \Delta p_{s,s+1}(k_s) \tag{21f}$$

where $k=0,1,\dots,N$ and N is the prediction horizon [10]. n is the number of vehicles in the platoon. $\vec{x}-\vec{x}_{ref}$ denotes the deviations from the desired positions and velocities [4]. The reason for having $\vec{x}-\vec{x}_{ref}$, is so that the entries in $\vec{x}$ approach the entries in $\vec{x}_{ref}$ instead of zero when the cost-function is minimized.

Equation (21b) denotes the state constraints of the system. Equation (21c), denotes the safety distance, i.e., inter-vehicular distances between any two vehicles can not be less than $1\,\text{m}$ (the $+L$ has been added for the same reason as in the reference vector $\vec{x}_{ref}$ in equation (2)). Equation (21d) and (21e) is the minimum and maximum constraints for velocity and acceleration respectively. Observe that the constraints (21b)-(21e) are all valid for all the time steps $k=0,1,\dots,N$.

The constraint in equation (21f) is the one that initiates the split. Assume here that the goal is to split between vehicle s and vehicle $s+1$. Thus, the distance between these vehicles, $\Delta p_{s,s+1}$, needs to be greater than or equal to $\Delta p_{ref,s,s+1}+L$ at time k_s. k_s denotes the time instant when the split should be done. It was observed that multiple splits got canceled because of a margin of $1\,\text{m}$. That's why the extra $+1$ has been added to equation (21f).

It is important to notice that the constraint in equation (21f) is only active under certain conditions. If these conditions are not met the constraint in equation (21f) is ignored. The first condition, $k_s<k+N$, checks if the split position, k_s, can be seen by MPC's prediction horizon. Assume that we are at time k, then the MPC can see until time step $k+N$, see figure 3. If the split position, is in this horizon, for example k_{s1} and k_{s2} are in this prediction horizon in figure 3, then the first condition is met. Since $k_{s3}>k+N$ the constraint in equation (21f) is ignored at time k. From this, we can deduce that the split will not start until the splitting point is in the prediction horizon. The other condition, $k<k_s-3$, makes so that the constraint is removed 3 steps before the splitting point. For example $k_{s1}<k+3$ in figure 3, therefore the constraint in equation (21f) is ignored. The reason for this is explained more in detail in section IV-B. Consequently, only k_{s2} makes the constraint in equation (21f) valid. Notice that, equation (21b) through (21f) are hard constraints and therefore, they have to be fulfilled. Thus, if the MPC can not accomplish the split under these hard constraints the split will be canceled.

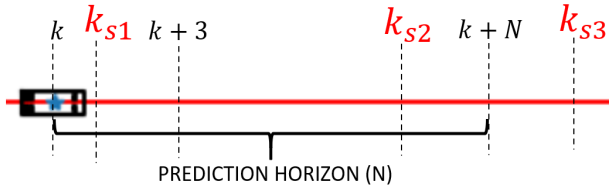


Fig. 3. Prediction horizon of a vehicle.

The cost-function $J(\vec{x}-\vec{x}_{ref},\vec{u})$, given in equation (21a), has the following form in our project [10]:

$$J(\vec{x}-\vec{x}_{ref},\vec{u})=\sum_{z=0}^{N-1}\Big(||\vec{x}(k+z|k)-\vec{x}_{ref}(k+z|k)||^2_{\mathbf{Q}}+ \\ +||\vec{u}(k+z|k)||^2_{\mathbf{R}}\Big)+||\vec{x}(k+N|k)-\vec{x}_{ref}(k+N|k)||^2_{\mathbf{Q}_N}, \tag{22}$$

where

$$||\vec{x}-\vec{x}_{ref}||^2_{\mathbf{Q}}=(\vec{x}-\vec{x}_{ref})^T\mathbf{Q}(\vec{x}-\vec{x}_{ref}). \tag{23}$$

At time k, $\vec{u}(k+z|k)$ is the predicted control signal, for time $k+z$. The same logic is also valid for the state vector $\vec{x}(k+z|k)$. Here $\mathbf{Q}$ and $\mathbf{R}$ are positive definite weight matrices for state and control variables respectively. The $\mathbf{Q}$ matrix contains the weights/costs associated with the deviation of $\vec{x}$ from $\vec{x}_{ref}$. The $\mathbf{R}$ matrix contains the weights/costs of $\vec{u}$ deviating from zero. The $\mathbf{Q}$ matrix is defined as

$$\mathbf{Q}^{(2n-1)\times(2n-1)}=\begin{bmatrix} q_1 & & & \\ & q_2 & & \\ & & \ddots & \\ & & & q_{2n-1} \end{bmatrix} \tag{24}$$

and the $\mathbf{R}$ matrix is defined as

$$\mathbf{R}^{(n-1)\times(n-1)}=\begin{bmatrix} r_1 & & & \\ & r_2 & & \\ & & \ddots & \\ & & & r_{n-1} \end{bmatrix}. \tag{25}$$

The constants q_i and r_i will be explained in section IV. $\mathbf{Q}_N$ is the terminal cost matrix and is equal to $\mathbf{Q}$ in this project, i.e. $\mathbf{Q}_N=\mathbf{Q}$ [10]. How the optimization problem is solved, equation (21a)-(21f), is outside the scope of this project, therefore an external tool in Python was used, namely the 'cvxpy' library.

When the optimization problem is solved, the MPC returns N predictions for $\vec{x}(k)$ and $\vec{u}(k)$, i.e. $\vec{x}(k+z|k)$ and $\vec{u}(k+z|k)$ for all $z\in 0,1,\dots,N$ are returned. Only the first predicted control vector, $\vec{u}(k+1|k)$, is applied at the next time step. The other $N-1$ predicted control vectors are discarded. All of the predicted $\vec{x}(k)$ for the future N steps are used in the next MPC calculation to obtain T_i, U_i and S_i. These values are used for one time step, Δt, and then the MPC solves the same optimization problem again but now, one time step later with a new initial condition.

IV. Method

In order to get accurate approximations for the discretized state-space representation and the linearized air drag, Δt has to be small. Therefore in this project, we chose this value to be $\Delta t=0.2\,\text{s}$. We chose the prediction horizon N to

be 30. Thus, the MPC can predict $N \cdot \Delta t = 6\,\mathrm{s}$ into the future. The prediction horizon was decided empirically. The computation time for the simulation scales proportionally with the prediction horizon N, so it can not be arbitrarily large. Making N too small, on the other hand, resulted in unsuccessful splits in some cases.

A. Implementing linearized air drag to the MPC

The constants used in this project for calculating the air drag are given in table I and most of the values are taken from [12]. With these values when a split is present, the air drag given in equation (12), is in the range of $[0.01, 0.07]\ \mathrm{m/s^2}$. When a split is not present the common value for the air drag is approximately $0.04\,\mathrm{m/s^2}$.

TABLE I
CONSTANTS THAT HAVE BEEN USED FOR CALCULATING THE AIR DRAG

Symbol	Value
ρ	$1.225\,\mathrm{kg/m^3}$
A	$6\,\mathrm{m^2}$
m	$40\,000\,\mathrm{kg}$
$C_{d,0}$	0.8
k_a	4.144
k_b	7.538

To make the MPC better at predicting the future states, the state-space representation should include the air drag experienced by the vehicles. As mentioned before, in this project, the air drag was linearized in the MPC calculations. The values T_i, S_i and U_i given in equation (14) are the values representing the air drag in the state-space representation. Since the air drag was approximated using a first-degree Taylor polynomial, it is only accurate for a small range of values around $\Delta p_{i,i+1}$ and v_i, used in the approximation. Because of this, the approximation had to be recalculated for each time step within the prediction horizon. The Taylor polynomial has to have values for $\Delta p_{i,i+1}$ and v_i to approximate around, but these future values are not known. A way around this is to use the predicted values for $\Delta p_{i,i+1}$ and v_i from the previous MPC calculation. This has also been shown in the Algorithm 1. Here we see that the matrix 'x_predicted' contains the predicted values from the currently calculated MPC, in line 24. Then this matrix is used in the next cycle at line 15 when the linearized air drag is being calculated (for calculating T_i, S_i and U_i). Observe that on line 14 we have the condition $i > 1$; this is so that the first calculation in the MPC uses the ideal state-space matrices given in equation (10). This is because the very first calculation done by the MPC does not have the chance to approximate $\Delta p_{i,i+1}$ and v_i with the predicted values from the last MPC calculation.

B. Hard and soft constraints

The biggest feature that distinguishes our project from earlier projects is that we put a hard constraint on the splitting distance at the splitting point when the splitting point is within the prediction horizon. This hard constraint was stated in equation (21f). When this hard constraint is not active, because of the conditions mentioned in equation (21f), the platoon is moving accordingly to the soft constraints in the $\vec{x}_{ref}$ vector.

Assume that we have a vehicle at time k such as in figure 3. In the next time step, this vehicle will be at time $k+1$. Since the split position, k_{s2}, is stationary, the vehicle has got 1 step closer to the splitting point. It is essential to consider this since the splitting point has decreased 1 step in the prediction horizon. This can also be seen in Algorithm 1 at line 7. Here we calculate which time step in the prediction horizon, the

Algorithm 1: A split algorithm for a platoon

```
1  Create a reference vector such as in equation (2);
2  Ask user for inputs;
3  i = 0;
4  while Split is not done & maximum time not exceeded do
5      i += 1;
6      if MPC sees the split point then
7          k_hc = Calculate which step in the prediction
             horizon the hard constraint corresponds;
8          if it is more than 3 time steps to the splitting
             point then
9              Set hard constraint to 'TRUE';
           else
10             Set hard constraint to 'FALSE';
11             Renew the reference vector;
           end
       end
12     Create ideal state space matrices as equation (6)
         and (8) ;
13     while k < N, k++ do
14         if lin. air drag added to the MPC & i > 1 then
15             To calculate lin. air drag use x_predicted(k);
16             Renew state-space according to equation
                 (19);
           end
17         Create the cost function according to equation
             (22);
18         Add constraints according to equation
             (21b)-(21e);
19         if hard constraint is 'TRUE' then
20             Set hard constraint on split distance as
                 equation (21f) and k_hc ;
           end
       end
21     Minimize the cost function and obtain control
         signals;
22     if control signals can not be obtained then
23         Split impossible, hence cancel the split. Set the
             reference vector to initial;
       end
24     Set x_predicted to the predicted state values for the
         N future values;
25     Update states;
   end
26 Plot graphs;
```

hard constraint corresponds to with

$$k_{hc} = \left\lceil \frac{t + N \cdot \Delta t - t_s}{\Delta t} \right\rceil, \tag{26}$$

where t is the current time, t_s is the splitting point given in time and $\lceil y \rceil$ represents that y is rounded up to the nearest whole number. This hard constraint is then used in the MPC calculation at line 20.

The first condition in equation (21f) is represented at line 6 in Algorithm 1. The second condition is represented at line 8 where it is stated 'if it is more than 3 time steps to the splitting point'. If both line 6 and 8 are fulfilled, the hard constraint is set to 'TRUE'. The reason for having the second condition is because having a hard constraint into the last time step resulted in unsuccessful splits. When the hard constraint was removed and the soft constraint was set, 3 steps before the splitting point, we observed that more splits could be accomplished. The soft constraints are set on line 11. Notice that until 3 time steps to the splitting point the soft constraint is not changed and is equal to the initial. The soft constraint is changed when the hard constraint is removed.

The soft constraint for the splitting distance is implemented by changing the reference from the initial distance to the desired distance in the $\vec{x}_{ref}$ (see the dotted line in figure 5 and 6). The other inter-vehicular distances are not changed. Even though the reference is set to the initial distance for so long, the only way to meet the hard constraint when solving the optimization problem is to deviate from the reference. However, since the MPC also wants to fulfill the soft constraint on the splitting distance at the same time, it will deviate from the reference as late as possible. This is because we want to utilize the benefits of the platoon for as long as possible.

C. Cost matrices

As mentioned in section III the $\mathbf{Q}$ and $\mathbf{R}$ matrices are weight matrices. These matrices are diagonal as can be seen in equation (24) and (25). The more important entries in the state vector $\vec{x}(t)$ and in the control vector $\vec{u}(t)$ have a greater weight corresponding to them. These weights describe the cost of deviating from the reference for the corresponding element in $\vec{x}(t)$ or $\vec{u}(t)$. See figure 4 for a demonstration of the weights when a platoon with six vehicles are present and the split is performed between vehicle 3 and 4. The basic weights for all the inter-vehicular distances and the velocities of the vehicles are 10 and 1, respectively. The weight for the inter-vehicular distance that will be split is 40 (q_3 in figure 4), this has the effect that the split will be performed faster at the disadvantage that the other inter-vehicular distances might deviate from the desired references momentarily. The inter-vehicular distance in front of the split distance has a weight of 30 (q_2 in figure 4) since it was observed that the distance between vehicle 2 and 3, in this case, often reached the minimum safety distance. The weights for all the control signals $\vec{u}(t)$ are 1 (r_1, r_4 and r_5 in figure 4), except for the two vehicles on either side of the split distance which have a weight of 10 (r_2 and r_3 in figure 4) corresponding to their control signals in the $\mathbf{R}$ matrix.

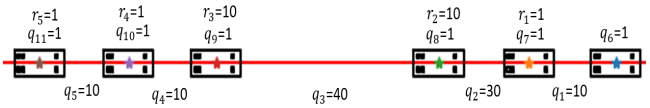


Fig. 4. The weight matrices' entries, when a split between vehicle 3 and 4 is present.

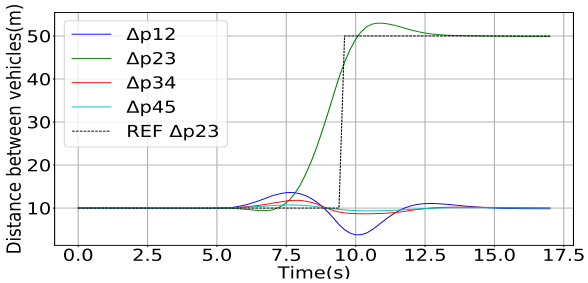


Fig. 5. Splitting between vehicle 2 and 3 up to 50 m at the time 10 s.

The $\mathbf{Q}$, $\mathbf{R}$ and the state-space matrices ($\mathbf{A}$, $\mathbf{B}$ and $\mathbf{D}$) are used when the cost function is minimized with respect to $\vec{u}(t)$. This minimization is beyond the scope of this project. Therefore, Python's built-in function 'cvxpy' has been used. The constructed MPC and the constraint are given to this solver and then this solver returns the optimal value for the split. If the solver can not obtain a solution, it returns 'None'. This means that the split is physically impossible under the given constraints. Then the split is canceled. To create the plots shown in section V, Python's built-in 'matplotlib' library has been used. The pseudo-code for the developed algorithm is given in Algorithm 1.

V. Results and Discussion

A. Successful splits up to 50 meters

The created system is capable of performing splits of up to 50 m between any two vehicles in a platoon of up to 10 vehicles. For larger split distances and longer platoons, sufficient experiments were not conducted to guarantee successful splits. In figure 5 the split between the second and third vehicle is shown and in figure 6 the split between the third and fourth vehicle is shown. In both of these scenarios, the split distances are 50 m, the platoon contains of 5 vehicles and the splits are set to be completed at the time 10 s. Since it is easier to perform a split for smaller splitting distances than 50 m, it is understood that these splits are also possible.

Notice that in figure 5 the distance at time $t = 10$ s is less than 50 m, precisely 49.25 m. Nevertheless, it is still considered as a successful split because of the 1 m margin, which has been added to the system. This means, for example if the goal is to reach 50 m at time $t = 10$ s then the distance should be more than 49 m at this time. This has been useful because it was observed that many splits got canceled just because they ended up 1 meter behind the target distance at the split time (which would be the case in figure 5). Since 1 m is a small distance, it can be ignored in this context.

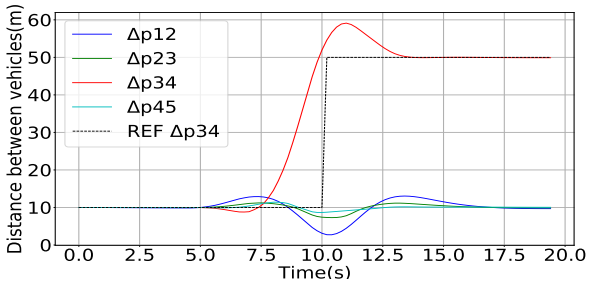


Fig. 6. Splitting between vehicle 3 and 4 up to 50 m at the time 10 s.

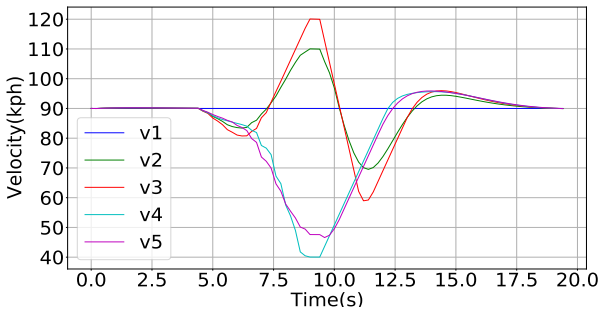


Fig. 7. The velocities of the vehicles when a split between vehicle 3 and 4 up to 50 m is present.

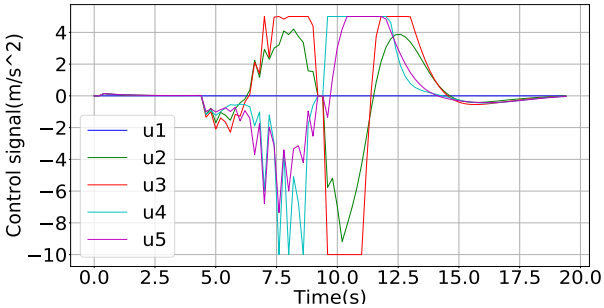


Fig. 8. The control signals of the vehicles when a split between vehicle 3 and 4 up to 50 m is present.

B. Big overshoot

From figure 5 and 6 it can be seen that the system behaves differently depending on between which vehicles the split is performed. For example the overshoot is much bigger in figure 6 than in figure 5. The overshoot, in figure 6, is around $20 - 25\%$. A big overshoot can affect the congestion on a road with heavy traffic. Attempts have been made to reduce the overshoot by changing the $\mathbf{Q}$ and $\mathbf{R}$ matrices in the cost function. In some scenarios, this decreased the overshoot, but other problems were observed, e.g., the split could not be accomplished for 50 m. Therefore, we can say that it is hard to optimize the MPC. Although, it is worth mentioning that the overshoot tends to increase with increased distances. When the same experiment as in figure 6 was repeated for different target distances, the overshoots increased with larger target distances, see table II. However, in some cases, this rule does not apply. For example, when the experiment in figure 5 was repeated for different target distances, the overshoots did not increase with the target distances, as can be seen in table III. The exact reason for the decrease in the overshoot when the target distance increases is unknown.

TABLE II
OVERSHOOTS WHEN SPLITTING BETWEEN VEHICLE 3 AND 4

Split distance	Overshoot
20 m	4.8%
30 m	6.5%
40 m	8%

TABLE III
OVERSHOOTS WHEN SPLITTING BETWEEN VEHICLE 2 AND 3

Split distance	Overshoot
20 m	4.0%
40 m	8.6%
50 m	7.5%

C. Constraints have been fulfilled

The velocities for each of the vehicles when the split is present between vehicle 3 and 4 (corresponding to figure 6) can be seen in figure 7. In the same scenario, the control signals for each vehicle can be seen in figure 8. In figure 7 we can see that the velocities for the different vehicles do not exceed the velocity constraints given in equation (21d). Similarly, from figure 8 we can see that the control signals do not exceed the constraint given in equation (21e). Observe that when the MPC demands higher/lower control signals, the control signals are saturated on the maximum/minimum possible value for a while. When the demanded value ends up within the constraint range, the control signal again fulfills the demanded value. The same thing can be observed with the velocities in figure 7. Consequently, while the split is happening, none of the constraints have been overruled. The same characteristic can be seen for the other splitting scenarios also. This is important because the constraints are based on physical conditions, governmental rules, etc. (see section II).

From figure 8 we see that the control signals, thus also the accelerations of the vehicles, are changing drastically over time. The control signals saturate at the borders of the constraints. Furthermore, when the control signal goes from -10 to $5\,\mathrm{m/s^2}$ the slope is extremely high. This leads to an uncomfortable ride for the passengers in the vehicles. Another thing that leads to an uncomfortable ride is the rapid changes in the control signals. For example, the purple and blue curves in figure 8 are almost like a 'sawtooth wave'. These increases and decreases will cause passengers to be thrown back and forth in the vehicle. Therefore, the extremely rapid changes in acceleration is a shortcoming of this system.

D. Impact of the hard constraint

As mentioned in section IV, the MPC can see the splitting point 6 s before it is reached. Thus, in the simulations shown in

figure 5 and 6 the split does not start before $t = 4\,\mathrm{s}$, since the splitting point is at $t = 10\,\mathrm{s}$. At time $t = 4\,\mathrm{s}$ the hard constraint is set for the splitting distance as mentioned in section IV. Even if it is hard to see that the split starts at $t = 4\,\mathrm{s}$ from figure 5 and 6 it is clear from figure 7 and 8 that the split starts at this time. On the other hand, from figure 5 and 6 it can be seen that the inter-vehicular distances are kept as small as possible for as long as possible, until $t = 7.5\,\mathrm{s}$ the inter-vehicular distances are kept small. This is good because then the vehicles can benefit more from the platoon's advantages.

Letting the split start when the prediction horizon can see the splitting point has some disadvantages as well. For example, a split that takes more than $6\,\mathrm{s}$ can not be accomplished. This is a drawback with using a hard constraint for deciding a split. Furthermore, when a limit is set on the splitting time, the interval between the constraint limits must be large. For example, the splitting vehicle needs to slow down to $40\,\mathrm{km/h}$ on a highway to reach a split of $50\,\mathrm{m}$ which is not ideal. Therefore, if stricter constraints are needed, a tactical layer that can make smarter decisions than just looking at the end of the prediction horizon is needed. This tactical layer can then obtain between which vehicles, at what time and how long before the splitting point the split should start. Of course, the prediction time can be increased by increasing the prediction horizon N or/and the time step Δt. Nevertheless, this leads to other problems. For example, increasing the prediction horizon makes the calculation time longer. Increasing the time step leads to worse discretization both in the state space representation and for the linearized air drag (see equations (14) and (20)).

E. Linearized air drag is not necessary

One of the most important results of this project is that the MPC manages the split without adding linearized air drag to the MPC calculations even though the real air drag is included in the system. This can be seen in appendix B, where the inter-vehicular distances are almost identical whether linearized air drag is added to the MPC or not. This is because of the feedback loop in the MPC. Thanks to this feedback loop MPC can self-compensate for the errors. Thus, an approximation of the air drag has proved unnecessary. Although, before giving up on this completely, some real-world experiments where more disturbances are present, such as rolling resistance and difference in road topography, should be examined. When a lot of other disturbances are present, the linearization of the air drag could be useful.

F. Unsuccessful splits

The developed system in this project can understand when a split is physically possible or not before the split even starts. For example, if the target split distance is given as $100\,\mathrm{m}$ then under the given constraints and time interval, this split is impossible. Therefore the split never starts. This is an important feature of the system. Because if a split would start and not reach the targeted value at the desired time, this could cause accidents. When the MPC understands that the split is not possible, it can inform the tactical layer about the situation and then the tactical layer can take necessary precautions. Since the linearized air drag is not exactly equal to the real air drag, sometimes a split starts and gets canceled after a while. This is because the MPC's prediction of the future was not exactly the same as the actual future. After a few time steps, the positions and circumstances have been changed and now the MPC understands that the split is impossible and then cancels it. When this happens, the vehicles should go back to the initial distances. This can be seen from figure 9. Here the

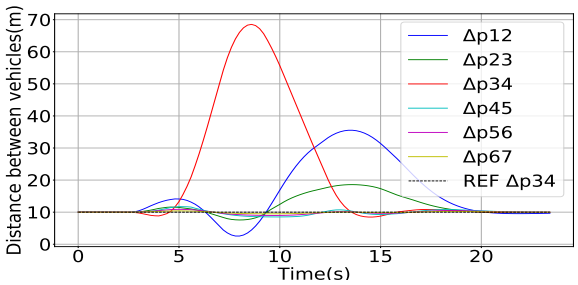


Fig. 9. Splitting between vehicle 3 and 4 up to $70\,\mathrm{m}$ at the time $8\,\mathrm{s}$ resulted in a unsuccessful split. The split was canceled.

goal was to create a gap between vehicle 3 and 4 up to $70\,\mathrm{m}$ at time $t = 8\,\mathrm{s}$. Unfortunately, the split got canceled too late, almost at time $t = 8\,\mathrm{s}$. This can lead to problems, as mentioned above. It was observed that the system behaved like this when the split distance was around $60\,\mathrm{m}$ - $70\,\mathrm{m}$. Distances above $80\,\mathrm{m}$ got canceled directly and distances under $50\,\mathrm{m}$ could be accomplished.

Before ending this section, we underline that the developed system can also achieve distances above $60\,\mathrm{m}$ in some cases. Since the goal was to split between any two of the vehicles up to $50\,\mathrm{m}$ and lack of experiments with longer distances, we settle with saying that the system only manages split distances up to $50\,\mathrm{m}$. Some split scenarios and whether they are accomplished or not, are given in the table IV. In this table linearized air drag has been added to the MPC. The

TABLE IV
SPLIT SCENARIOS WHEN THE SPLIT DISTANCE IS LARGER THAN $50\,\mathrm{m}$

# of vehicles	Split between	Split distance (m)	Accomplished
5	1 and 2	60	YES
2	1 and 2	60	YES
7	5 and 6	65	NO
3	2 and 3	65	NO
4	2 and 3	70	NO
5	3 and 4	70	NO

experiments and results (and far more) can be reproduced by the published code at [14].

VI. FUTURE STUDIES

In this project, the focus has been on splitting two vehicles in a platoon to a target distance at a given time. In general, a split is performed for a vehicle to merge into the created gap. Unfortunately, no vehicles could be merged in this project due to lack of time. Therefore, an important future study is

to merge a vehicle into the created gap and observe how the system reacts in different situations. Furthermore, the platoon needs a user to give commands where and when the split should occur in this project. This is not an optimal solution when a merging scenario is present. In this case, the platoon should sense by itself or get information from a tactical layer where the merging vehicle is. Moreover, the platoon itself (by an advanced tactical layer) should understand where and when a split should begin. Thus, an advanced tactical layer should be created. For this, the code in [14] can be further developed.

In this project, only the air drag has been taken into account. Resistances such as gravitational force when driving on a slope and rolling resistance have been ignored. Thus, a future study would be to add these resistances and observe how the system would react in these cases. When successful platooning and splitting are obtained, the project can be taken one step further. The algorithm can be tested in the real world. To do this, the code in [14] needs to be extended so that the vehicles can wireless communicate with a tactical layer.

In this project, a time where the split should be accomplished is given. This can be problematic in real-world applications. Because in the real world, a platoon should know $where$ and $when$ the split should be done. For example, the position of a ramp or a lane reduction should be given to the platoon so that the platoon should be done with the split at these positions.

One of the most significant benefits of platooning is fuel reduction. Therefore, the fuel consumption of a platoon can be calculated by using the fuel model represented in [2] and [15]. This can then be compared with the fuel consumption where the vehicles are not in the platoon. Furthermore, the waste of fuel when a split cancels can then also be calculated.

VII. Conclusion

Our goal with this project was to develop a system that can split a platoon. To achieve this, we used MPC to control the system because it handles constraints well and control systems with multiple variables. We showed that the constructed system was able to split between any two adjacent vehicles in a platoon up to a distance of $50\,\mathrm{m}$. Although the splitting was successful, a big overshoot was observed for some splits. Changing the cost matrices and the constraints is a way to decrease the overshoot, but it resulted in some splits not being completed. Linearized air drag was successfully added to the MPC, but the MPC could compensate for the air drag without adding linearized air drag to the MPC. MPC is useful for controlling a platoon and it is good at compensating for disturbances but can be challenging to optimize.

Appendix A
Complete expression for acceleration from linearized air drag

Appendix B
Extra plots

Acknowledgment

The authors would like to thank our supervisors Miguel Aguiar and Xiao Chen as well as the context responsible Jonas Mårtensson for their guidance and helpful feedback during the project. We would also like to thank Atsushi Sakai for his github code titled 'model predictive speed and steer control' which was very useful when learning how to implement a MPC in Python.

References

[1] A. Alam, "Fuel-efficient distributed control for heavy duty vehicle platooning," Ph.D. dissertation, KTH Royal Institute of Technology, Stockholm, 2011.

[2] V. Turri, "Fuel-efficient and safe heavy-duty vehicle platooning through look-ahead control," Ph.D. dissertation, KTH Royal Institute of Technology, Stockholm, 2015.

[3] W.-H. Hucho, "Aerodynamics of road vehicles," *SAE International*, 1986.

[4] W. Levine and M. Athans, "On the optimal error regulation of a string of moving vehicles," *IEEE Transactions on Automatic Control*, vol. 11, no. 3, pp. 355–361, 1966.

[5] A. Duret, M. Wang, and A. Ladino, "A hierarchical approach for splitting truck platoons near network discontinuities," *Transportation Research Procedia*, vol. 38, pp. 627–646, 2019.

[6] A. Sakai, D. Ingram, J. Dinius, K. Chawla, A. Raffin, and A. Paques, "Pythonrobotics: a python code collection of robotics algorithms," *arXiv preprint arXiv:1808.10703*, 2018.

[7] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: an operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020. [Online]. Available: https://doi.org/10.1007/s12532-020-00179-2

[8] S. Dasgupta, V. Raghuraman, A. Choudhury, T. N. Teja, and J. Dauwels, "Merging and splitting maneuver of platoons by means of a novel pid controller," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2017, pp. 1–8.

[9] A. Al Alam, A. Gattami, and K. H. Johansson, "Suboptimal decentralized controller design for chain structures: Applications to vehicle formations," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE, 2011, pp. 6894–6900.

[10] M. Cannon, "C21 model predictive control," 2016.

[11] Infrastrukturdepartementet, "Trafikförordning (1998:1276) 4 kap. 17 §," 1998. [Online]. Available: https://www.riksdagen.se/sv/dokument-lagar/dokument/svensk-forfattningssamling/trafikforordning-19981276_sfs-1998-1276

[12] D. R. Lopes and S. A. Evangelou, "Energy savings from an eco-cooperative adaptive cruise control: a bev platoon investigation," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 4160–4167.

[13] G. Sivaraj, K. Parammasivam, and G. Suganya, "Reduction of aerodynamic drag force for reducing fuel consumption in road vehicle using basebleed," *Journal of Applied Fluid Mechanics*, vol. 11, no. 6, pp. 1489–1495, 2018.

[14] E. Vardar and A. Gustafsson, "Splitting a Platoon Using Model Predictive Control," April 2021. [Online]. Available: https://github.com/Vardar98/kex/

[15] V. Turri, B. Besselink, and K. H. Johansson, "Cooperative look-ahead control for fuel-efficient and safe heavy-duty vehicle platooning," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 1, pp. 12–28, 2017.

CONTEXT B
AUTONOMOUS ROBOTIC SYSTEMS

POPULAR DESCRIPTION

Skip walking, the driver isn't talking

The introvert's paradise and the extrovert's nightmare: next time when you are ride-hailing there might not be a driver in the car, at least not a human. Our rides around town are gradually becoming more autonomous since it will be both cheaper and safer that way.

A non-human at hold of the wheel has several benefits. Not only will your wallet be happier since there is no driver that has to be paid, you will also save time by arriving at your destination earlier while simultaneously having more time by yourself. In addition, driverless taxis will be able to smoothly pick the most optimal route depending on traffic.

Safety is another considerable advantage. Imagine hailing a ride to go home late a Friday night and an autonomous taxi pulls up to take you home. The vehicle would safely take you to your destination without you needing to worry about your driver's intentions. Awkward conversations are also no more than a fading memory. With the help of sensors, cameras and real-time traffic updates, your ride will take you safely to wherever you want to go while taking the safest route and preventing accidents. This is another example of something that can not be guaranteed with a human driver. The autonomous vehicle will look after you and others without shouting angry words at other motorists and pedestrians.

The development of autonomous systems is moving at a fast pace. They are already deeply integrated into our transportation systems, and it will not be long until fully autonomous vehicles will cruise the streets. One key technology behind autonomous cars is neural networks, which has similarities to brain cells, with traffic awareness. To achieve a good neural network a lot of training is needed, just as when you're learning how to drive for the first time. This can be done with detailed and large amounts of information on how humans drive. When a software that is safer than humans there will not be a long time until robot taxis are commonplace.

SUMMARY OF PROJECT RESULTS

The science around autonomous systems is a rapidly evolving area. Repetitive tasks are continuously being replaced by autonomous systems to free up the human mind and thus increasing productivity and creativity. This can be seen from the inside of a car factory and all the way into our homes as vacuum cleaners. With increased control of complex dynamics and flexibility the autonomous robots will continue to expand their presence in the global society. The purpose of autonomous robots is to be able to perform without human interference, thus being capable of operating in complex environments and handling unforeseen events.

Three possible applications for autonomous robotic systems were developed by the project groups B1, B2, and B3. The goal of project B1 was to design a motion planning algorithm able to manage aggressive maneuvering and avoid obstacles. Project group B2 studied how self-driving vehicles in a formation could safely maneuver to a destination. In project B3, a motion plan was developed for robots working in a warehouse-like environment, making the agents perform specified timed tasks.

Project group B1 investigated the problem of creating a motion planning algorithm for a quadrotor UAV. It should be capable of creating a trajectory between an initial state and a goal state in an obstacle-cluttered environment and be able to manage aggressive maneuvering when needed. This trajectory should consider the shape of the UAV for the purpose of being dynamically feasible, limited by a max thrust, collision free and optimal by minimizing a cost function. To generate an optimal trajectory with respect to a cost in jerk, an extended variant of the rapidly exploring tree algorithm (RRT) called RRT* was used. To ensure collision avoidance and be able to manage aggressive maneuvers the group used a control barrier function

based method (CBF). The methods were tested and analysed in simulations for various environments, by plotting the trajectory with different obstacles in Python.

A future improvement of the project would be to take the constraints of an actual quadrotor into account when generating the trajectory. Furthermore, one could upgrade the obstacle detection algorithm used in the CBF algorithm to make it more computationally efficient and by taking the shape of the UAV into account in a more effective way.

Project group B2 explored ways of safely moving a formation of ground vehicles from a starting position to a final destination with static obstacles. This was accomplished by implementing algorithms for three of the following areas: trajectory tracking, formation control and collision avoidance. In essence, a centralized system receives information from the vehicles, performs calculations and sends instructions back. Most algorithms that are used here, use methods from control theory in order to calculate the general movement of each vehicle. To verify the performance of these methods; stability analysis and simulations in Matlab were done.

To further expand on this, future projects could investigate the idea of moving obstacles, different formations and different numbers of vehicles with various shapes. In addition, one could look at ways to find an appropriate formation given different shapes and numbers of vehicles.

In B3, the project group created motion plans for two robots to move in a formation towards a goal area. The robots were assigned timed tasks, created to be completed within a specific time interval, simultaneously avoiding collision with the other robot. The mathematical language Signal Temporal Logic (STL) was used to create and specify the tasks, making every task valid only during the specified time interval. A control barrier function (CBF) based algorithm was then used for control design to specify the route, which the robot would take to complete the given task. The motion plan algorithm was coded in Matlab by the group and tested through simulations inspired by a warehouse environment, where the robots move goods in a formation.

A future development within the area of project B3 could be to further expand the robots' ability to navigate and complete tasks in an environment more resembling a warehouse in detail. An example of this would be adding obstacles within the workspace.

IMPACT ON SOCIETY AND ENVIRONMENT

The fast emerging technology of autonomous robotic systems will make its presence increasingly common in our lives and the society for various purposes. Considering the wide uses this technology implicates, it is important to contemplate and understand how autonomous technology could affect us in our daily lives and how it could impact the environment. As with most great innovations, there will be both ethical and social challenges to solve.

The labor market is a main societal part that is already affected by autonomous systems. Robots that are replacing workers will have an impact on the economy due to the decreasing number of taxpayers and thus also the tax revenue will decrease. Companies will most likely want to hire autonomous robots instead of humans since they do not have to pay taxes for social security for the robots. In order to keep the economy in balance and reduce the unemployed rate, a shifting of the tax burden from labor to capital is required. The robot taxes should also variate, e.g. companies with high profits that depend a lot on autonomous robots should pay higher taxes.

To add on the tax issue, autonomous systems also have a tendency to replace low entry jobs because these are often repetitive jobs that do not require high specialization. Many industrial jobs have already been replaced by autonomous systems but over time other low entry jobs will also be replaced. Although new jobs are created, they tend to require a higher educational qualification, since these new jobs are often related to the technology that replaced the previous jobs, e.g, maintenance and repair. As a result, the labor market's entry barriers are raised. This makes it more difficult for low educated individuals to have employment, subsequently that may lead to increased unemployment rates. For instance, young people often rely on low entry jobs to achieve financial independence and experience. Higher entry barriers would require a longer education period that the government needs to be ready to finance, to make it sustainable for the individual that is more or

less forced to follow that path. As this path is more technology focused, it may lessen the diverse occupation options for the individual. But if society continues in this direction, the average level of education would further increase. This may lead to increased possibilities of technological advancements, benefiting society as a whole.

The question of integrity regarding autonomous systems is another important point. Autonomous robots have a high capability of collecting and processing information, which will inevitably contain data about people's private lives and habits. An unmanned robot might, for example, carry a camera to assist it in navigating different areas, and could save this data for future usage or reference. This video footage could possibly show citizens and their location at certain moments. A cell phone could track its own position to display accurate weather conditions, but will at the same time reveal its owner's location and movement patterns. Some robots may in other ways obtain sensitive information regarding one's integrity, for example healthcare robots, where heart rate, blood pressure or blood sugar levels can be measured. While all this information could be used to detect and prevent threats or health issues, it could also be subject to data leaks, cyber attacks or in other ways fall into the wrong hands. If this were to happen, the individual's right for privacy would be at risk.

Who is to decide the amount of information that should be allowed to be collected? The first thing should be to prevent the autonomous robots from collecting any unnecessary information on the hardware stage. If a robot can manage its tasks with sensors instead of a camera, this should be the prioritized solution. Furthermore, legislation is necessary to regulate what kind of data is allowed to be stored. It is clear that the right precautions and methods have to be used to ensure privacy for every citizen. If this can be done, autonomous robots have the potential to improve the safety and well-being in the whole of society.

The environmental impact of autonomous robotic systems depends on their primary purpose. One common aspect is the rise of production-based emissions that an increase in autonomous systems will have. These are caused by the materials needed, including the electronics that the robot consists of. However, this impact could be decreased if the materials that are mainly used for producing robots in mass production are locally sourced and are easy to recycle. When autonomous robots are replacing human workers, the energy consumption will increase. Another risk with an increased use of autonomous robots in production would be that the total use of energy and material will increase as the unit cost decreases.

However, even when considering these environmental costs, we still believe that autonomous robots have a great potential in reducing the impact humans have on the environment.

Primarily, autonomous systems have the potential to make several processes more energy efficient than if they are done by humans. Transportation is an example of this, where autonomous vehicles can reduce the energy cost by driving smoother with fewer decelerations and accelerations than human drivers. Autonomous technology could also be used in machines to detect and respond intelligently to changes in their environment, so that only the energy and resources needed are used. This can significantly reduce the resources required in everything from the manufacturing industry to public transport. An example could be maintenance robots that can detect when solar plants, wind farms and power grids need service.

Furthermore, the ecology can also benefit from autonomous robots as these can be used to monitor the quality of the environment in areas that are inaccessible for humans in order to increase the awareness of the problems we are facing. A clear example of this is that drones can be used to detect methane leaks from old wells.

Motion Planning for Aggressive Flights of an Unmanned Aerial Vehicle

Alexander Medén and Erik Warberg

***Abstract*—Autonomous Unmanned Aerial Vehicles (UAV) have great potential in executing various complex tasks due to their flexibility and relatively small size. The aim of this paper is to develop a motion planner capable of generating a trajectory with aggressive maneuvers through narrow spaces without collision. The approach utilizes a framework using an optimized variant of the Rapidly-exploring Random Tree (RRT) algorithm, called RRT*, with a Control Barrier Functions (CBF) based obstacle avoidance algorithm as well as a motion primitive generator. If a motion primitive collides with an obstacle, the obstacle avoidance algorithm will attempt to reach the end state of a motion primitive in a collision free manner while complying with the actuation constraints. From the collision free trajectories an optimal path is continuously searched for by RRT* by minimizing a cost in jerk. The performance of RRT* and the obstacle avoidance are tested in simulations independently and jointly, in several different scenarios. The resulting motion planner successfully finds a high-level trajectory for the different scenarios. Limitations of the method as well as possible areas of improvements are also discussed at the end of this paper.**

***Sammanfattning*—Autonoma UAV har goda möjligheter för att utföra flera olika komplexa uppgifter tack vare deras flexibilitet och storlek. Denna rapport redogör för en rörelseplaneringsalgoritm som kombinerar manövrerbarheten hos en UAV för att skapa en kollisionsfri bana som innehåller aggressiva manövreringar genom trånga utrymmen. Tillvägagångssättet innefattar att kombinera *Rapidly-exploring Random Tree* (RRT*) med en algoritm för att undvika hinder baserad på *Control Barrier Functions* (CBF), samt att låta banan delas upp i segment, så kallade *motion primitives*, som genereras var för sig. Om en *motion primitive* kolliderar kommer den hinderundvikande algoritmen göra ett försök att nå dess målposition medan kollision undviks och manövreringsbegränsningarna uppfylls. Med en samling genomförbara *motion primitives* söker RRT* efter en kontinuerlig bana optimerad med hänsyn till en kostnad i ryck. Prestandan för RRT* och den hinderundvikande algoritmen simuleras både separat och tillsammans. Den resulterande rörelseplaneraren lyckas hitta en genomförbar bana för vardera scenario. Begränsningar av metoden samt potentiella förbättringsområden diskuteras i slutet av denna rapport.**

***Index Terms*—UAV, obstacle avoidance, RRT, motion planning, aggressive maneuver, control barrier functions.**

Supervisor: Xiao Tan

TRITA number: TRITA-EECS-EX-2021:141

I. Introduction

An UAV refers to a type of aircraft, capable of operating without the need for a human pilot. There are five broader groups of UAVs according to Sebbane [1] of which this project is aimed to the smallest group known as Mini/Micro Tactical UAV, specifically a quadrotor UAV. There are various applications for UAVs such as different kinds of surveillance and data retrieval, as well as emergency operations [2], [3]. There has recently been extensive development of different kinds of technologies allowing the use of UAVs for various purposes [4]. On the other hand, in [5, Introduction] it is explained that autonomous UAVs still require substantial development in both technological and mathematical areas in order to become truly autonomous, for instance obstacle avoidance, collision detection and execution of complex maneuvers.

For an UAV to be able of reaching a desired destination a motion planner is necessary. The planner needs to find a feasible trajectory, defined as the path of the UAV, that considers both the dynamics of the UAV as well as the obstacles in its environment. To overcome these problems there has been plenty of previous research on search-based high-level motion planning algorithms, such as A* and RRT among others, summarized by Aggarwal and Kumar [6]. These can be combined with low-level trajectory generators such as [7] and [8] to create trajectory segments, called *motion primitives*. However, these motion primitives do not account for obstacles.

In this paper we aim to create a motion planner for a quadrotor UAV that combines an optimized variant of RRT, called RRT*, with an obstacle avoidance algorithm that generates motion primitives that do account for the obstacles. By using this combination the planner is expected to be able of finding a feasible trajectory from an initial state to a goal state, called a *high-level trajectory*, in an obstacle-cluttered environment. The trajectory should also be optimized by a cost in jerk as well as being able of managing aggressive maneuvers. An aggressive maneuver is defined as flying with high inclination and close to obstacles, such as through a narrow vertical gap formed by two thin walls. Two critical parts to succeed with such maneuvers are a good dynamic model and an obstacle avoidance algorithm with consideration of the shape of the UAV. With aggressive maneuvering it is possible to increase the capabilities of the UAV to a wider array of tasks.

The RRT* algorithm generates a high-level trajectory by using a graph represented by a set of vertices, which are the endpoints of the motion primitives, and a set of edges, which are the motion primitives connecting the vertices. The algorithm expands the graph from a root vertex by generating vertices containing a random attitude and a random position of the UAV. By randomizing the attitude and the position in RRT*, there is a great variance of trajectories assembled to generate the high-level trajectory, which is favourable when optimizing by a cost in jerk. When a collision is detected in a motion primitive, the obstacle avoidance is activated. The obstacle avoidance uses Control Barrier Functions that

considers the barriers of the obstacles as well as the UAV, to define a safety set. By generating a control signal which follows a set of safety constraints, the obstacle avoidance algorithm can produce a motion primitive that assures safety for the UAV.

To simulate the final high-level trajectory generated by the motion planner a simulation environment in Matlab is used from the work of [9], illustrated in Fig. 6.

The succeeding parts of this report is structured into six sections. Starting with section II, we describe the dynamic model of a quadrotor. In section III the problem statement is found, followed by section IV, that describes the method to solve the problem. The results are presented in section V and discussed in section VI. Lastly, the conclusions of the project are given in section VII.

II. Dynamic Model

To describe the state of the UAV it is necessary to have a dynamic model. This model relates position, velocity, acceleration, rotation as well as angular velocity and angular acceleration to the four rotor speeds.

A. Navigation

Two coordinate systems are used, a world frame W and a UAV body-fixed frame B, as introduced in [8]. The rotation of B is defined by the Euler angles denoted as yaw, pitch and roll (ϕ, θ, ψ) in order to construct a rotational matrix R. This matrix describes the rotation of B in relation to W. This relation in necessary since the thrust f_i from each rotor is in the z_B direction, as seen in Fig. 1.

B. UAV Characteristics

By using the two reference frames it is now possible to define the different variables of the UAV relevant to the dynamics, as done in [10].

$m \in \mathbb{R}$	the total mass of the UAV
$J \in \mathbb{R}^{3\times3}$	the inertial matrix with respect to B
$R \in SO(3)$	the rotational matrix from B to W
$\Omega \in \mathbb{R}^3$	the angular velocity in B
$\omega_i \in \mathbb{R}^3$	the angular velocity in B of the ith rotor
$x \in \mathbb{R}^3$	the position vector of the center of mass in W
$v \in \mathbb{R}^3$	the velocity vector of the center of mass in W
$f_i \in \mathbb{R}$	the thrust generated by the ith rotor along $\vec{z}_B$
$f \in \mathbb{R}$	the total thrust magnitude
$L \in \mathbb{R}$	the distance between the center of mass and each rotor.
$M \in \mathbb{R}^3$	the total moment vector in B

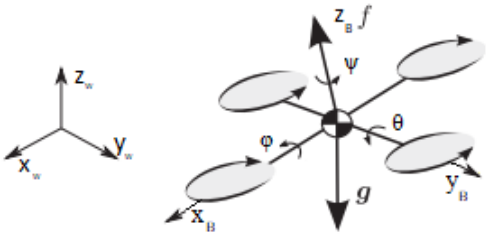


Fig. 1. Illustration of the two coordinate systems, modified from [8].

C. Control and Dynamics

The actuators of the UAV, which runs the four rotors, require a control signal u to produce the desired thrust and moment, as defined in (1), where k_F and k_M are aerodynamic constants, which depend on the design of the rotors.

$$u = \begin{bmatrix} f \\ M_1 \\ M_2 \\ M_3 \end{bmatrix} = \begin{bmatrix} k_F & k_F & k_F & k_F \\ 0 & k_F L & 0 & -k_F L \\ -k_F L & 0 & k_F L & 0 \\ -k_M & k_M & -k_M & k_M \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{1}$$

With this setup, where the thrust and moment depends on ω_i, it is easy to allow for constraints on the actuators of the UAV.

With the equations (2) to (5) it is possible to describe the whole dynamics of a quadrotor and determine the state of the UAV from the properties of u.

$$\dot{x} = v \tag{2}$$

$$m\dot{v} = fRz_W - mgz_W \tag{3}$$

$$\dot{R} = R[\Omega]_\times \tag{4}$$

$$J\dot{\Omega} + \Omega \times J\Omega = M \tag{5}$$

The notation $[.]_\times$ corresponds to the hat operator, which maps $\mathbb{R}^3 \to \mathbb{R}^{3\times3}$.

The state of the dynamical system can be described by the state vector

$$X = [x^T, v^T, \Omega^T, vec(R)^T]^T, \tag{6}$$

where $vec(.) : \mathbb{R}^{n\times m} \to \mathbb{R}^{(nm)\times1}$ is the row-wise vectorization.

From the relations described in (2) to (5) it is possible to express the dynamical system as in (7) with $\mathfrak{f} \in \mathbb{R}^{18\times1}$ and $\mathfrak{g} \in \mathbb{R}^{18\times4}$, which is the state-space model of the UAV.

$$\dot{X} = \mathfrak{f}(X) + \mathfrak{g}(X) \cdot u \tag{7}$$

III. Problem Formulation

The problem consists of creating a motion planning algorithm for a quadrotor UAV that is able to manage aggressive maneuvers in complex environments. The motion planner is expected to:

- create a feasible trajectory G_f between an initial state $X_0 = [x_0^T, v_0^T, \Omega_0^T, vec(R_0)^T]$ and a goal state $X_g = [x_g^T, v_g^T, \Omega_g^T, vec(R_g)^T]$ in an obstacle-cluttered environment,
- take the shape of the UAV into consideration,
- optimize the trajectory by minimizing a cost in jerk.

IV. METHOD

The approach to solve the problem formulated in section III is divided into four parts: *High-level Trajectory Generation*, *Collision Detection*, *Obstacle Avoidance* and *Controllers*.

In the *High-level Trajectory Generation* section it is explained how the motion primitives are generated and then assembled into a collision free high-level trajectory with RRT*. When checking the motion primitives for collision a method explained in the *Collision Detection* section is used, and if there is a collision, the algorithm in *Obstacle Avoidance* attempts safely steering. To steer the UAV to a desired position when the obstacle avoidance is activated, a position based flight mode controller, explained in *Controllers*, is used. To finally simulate the motion planner, the computed trajectory is tracked with a velocity and attitude based flight mode controller, also described in section *Controllers*.

A. High-level Trajectory Generation

G_f is defined as a collection of n concatenated feasible motion primitives $\sigma_{f,i}(t)$ in order. Therefore, G_f is expressed as

$$G_f = \{\sigma_{f,1}(t), \sigma_{f,2}(t), ..., \sigma_{f,n}(t)\}. \tag{8}$$

In order to make the optimization of G_f possible, each of the motion primitives are associated with a cost in jerk, denoted as J_σ in this paper, which is defined in [7, Eq. 13] and calculated as

$$J_\sigma = \frac{1}{T}\int_0^T \|\mathrm{j}(t)\|^2 dt, \tag{9}$$

where $\mathrm{j} = \dddot{v}$ is the jerk and T is the duration of the motion primitive.

1) RRT*:

The RRT* algorithm is used to find G_f, as well as optimize it. The optimization is done through the whole trajectory generation process with respect to the sum of the cost $J_{\sigma,i,j}$ for each motion primitive $\sigma_{i,j}$ at depth i and width j, defined later in IV-A2. The pseudo code for this algorithm is provided at the end of this subsection.

The trajectories generated by RRT* are stored in the graph $G = (\mathcal{V}, \mathcal{E})$, containing a set of vertices $\mathcal{V}$ and a set of edges $\mathcal{E}$, as in [11]. The graph is initialized with two vertices q_{root} and q_{goal}, containing the initial state and the goal state of G_f, respectively.

The RRT algorithm was introduced by LaValle in [12], and finds G_f in G by expanding a graph from q_{root} by generating K vertices containing a random position x_{rand} and a random attitude R_{rand}, within chosen intervals. As visualised in Fig. 2, the algorithm searches for the nearest vertex, $q_{nearest} \in \mathcal{V}$ when a new random vertex, q_{rand}, is generated. If the distance from $q_{nearest}$ to q_{rand} is greater than the incremental distance Δq, the new configuration q_{new} is determined by decreasing the distance to equal Δq, in the direction from $q_{nearest}$ to q_{rand}. If q_{new} does not coincide with an obstacle and there is no collision, checked for with the method described in subsection IV-B, in the motion primitive between the two vertices, then q_{new} is connected to $q_{nearest}$.

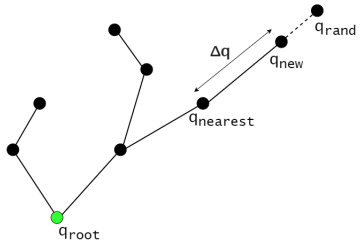


Fig. 2. Illustration of the RRT algorithm.

RRT is sufficient for the purpose of finding a feasible trajectory. However, it is not guaranteed that it will find an optimal trajectory. That is why the optimized variant, RRT*, introduced by Karaman and Frazzoli in [13], is used instead. RRT* is assured to find an optimal trajectory as the number of vertices approaches infinity.

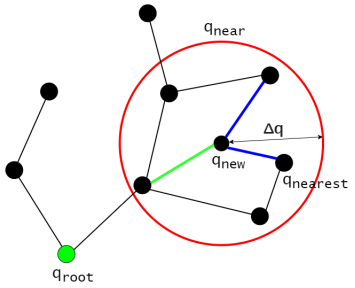


Fig. 3. Illustration of the RRT* algorithm.

A difference in RRT* from RRT is that a new configuration q_{new} does not necessarily connect to $q_{nearest}$. Instead, the algorithm checks for vertices q_{near} in $\mathcal{V}$ within a radius Δq from q_{new}, corresponding to the vertices inside the red circle in Fig. 3. The cost at the vertices in q_{near} are compared and q_{new} connects to the vertex with the lowest cost, corresponding to the green line in Fig. 3.

The cost of a vertex $q_{i,j}$, when $q_{0,0} = q_{root}$, is defined as

$$J_{\Sigma,i,j} = \sum_{n=1}^{i} J_{\sigma,n,k}, \quad k \geq 0. \tag{10}$$

Another difference RRT* has from RRT is the possibility of rewiring the vertices. When q_{new} has been connected to the cheapest parent vertex, all the remaining vertices in q_{near} are checked whether their cost would decrease if rewired to q_{new} instead of their current parents. This feature corresponds to the blue lines in Fig. 3. In this way RRT* is guaranteed to find G_f while at the same time minimizing its cost.

RRT* requires the following functions:

- **rand_conf**(): Returns a random vertex q_{rand}, containing a random x_{rand} and a random R_{rand}.
- **nearest_vertex**(q_{rand}, G): Returns $q_{nearest}$, which is determined by searching G for the nearest vertex by x.
- **new_conf**($q_{nearest}$, q_{rand}, Δq): Returns a new configuration q_{new}.
- **obstacle_free**(q_{new}): Checks if the vertex q_{new} coincides with an obstacle.

- **near_vertices(**q_{new}**,** Δq**,** G**)**: Returns a list of vertices q_{near} that are positioned within a radius Δq from q_{new}.
- **choose_parent(**q_{near}**,** q_{new}**)**: Returns the cheapest parent vertex q_{parent} to q_{new}.
- **rewire(**q_{new}**,** q_{parent}**,** q_{near}**,** G**)**: Rewires the vertices in q_{near} to q_{new} if that would decrease their cost.
- **A collision detection function**: Checks if there is a collision with an obstacle in a motion primitive between two vertices. The approach for this is explained in subsection IV-B.

Algorithm 1 RRT*

Input: incremental distance Δq, root vertex q_{root}, goal vertex q_{goal}
Output: feasible and optimized trajectory G_f

$G.init(q_{root}, q_{goal})$
for k = 1 to K **do**
  $q_{rand} \leftarrow rand_conf()$
  $q_{nearest} \leftarrow nearest_vertex(q_{rand}, G)$
  $q_{new} \leftarrow new_conf(q_{nearest}, q_{rand}, \Delta q)$
  if $obstacle_free(q_{new})$ **then**
    $q_{near} \leftarrow near_vertices(q_{new}, \Delta q, G)$
    $dist2goal = norm(q_{new} - q_{goal})$
    if $dist2goal \leq \Delta q$ **then**
      $connect2goal(q_{new}, G)$
    end if
    $q_{parent} \leftarrow choose_parent(q_{near}, q_{new})$
    $rewire(q_{new}, q_{parent}, q_{near}, G)$
  end if
end for
return G_f

2) Motion Primitive Generation:

Motion primitives are used in this paper, as they are allowed to be modified without changing other motion primitives in the high-level trajectory. This feature is essential for the ability of cost minimization, as the motion primitives can in this way be replaced by cheaper motion primitives in RRT*.

A motion primitive is denoted as

$$\sigma(t) = \sigma(x(t), v(t), \Omega(t), R(t)), \quad t \in [0, T], \tag{11}$$

and is always generated between two vertices, a start vertex q_{start} and an end vertex q_{end}, in RRT*. These two vertices are associated with the intermediate states $\sigma_{start} = \sigma(0)$ and $\sigma_{end} = \sigma(T)$ of a motion primitive, respectively. These motion primitives are primarily generated by using the work of [7] with the code available at [14]. But in case of a collision in a trajectory generated by [14], a motion primitive can instead be generated by the obstacle avoidance algorithm described in subsection IV-C in order to attempt reaching σ_{end}.

When generating a motion primitive with the method from [7], it is required to have a desired duration T and the translational variables position, velocity and acceleration for both $t = 0$ $(x_{start}, v_{start}, \dot{v}_{start})$ and $t = T$ $(x_{end}, v_{end}, \dot{v}_{end})$. Firstly, the end translational variables are set: $x_{end} = x_{rand}$, v_{end} is free for the algorithm to determine and $\dot{v}_{end} = k_s R_{rand} z_W$. The product $R_{rand} z_W$ corresponds to the normal vector of the UAV and the constant $k_s \in (0, 1]$ is used to scale $\dot{v}_{end}$ to minimize the jerk of the trajectory. Therefore it is desirable to set $||\dot{v}_{end}||$ of a motion primitive to be near zero in magnitude, suggesting that k_s should be small. The purpose of this is to counteract oscillatory behaviour but it cannot be zero in order to still be able to describe the attitude of the UAV at σ_{end}.

Since σ_{start} is defined from a previous motion primitive as in (12) the transition between motion primitives is continuous.

$$\sigma_{i,j}(0) = \sigma_{i-1,k}(T_{i-1,k}), \quad i > 0, \;\; j, k \geq 0 \tag{12}$$

Therefore, the start translational variables are inherited from the end translational variables of the previous motion primitive, unless $\sigma(0) = X_0$ for which $||\dot{v}|| = 0$. If the previous motion primitive is generated by the obstacle avoidance algorithm, the start translational variables are calculated with the use of *differential flatness*, explained in [8].

The motion primitive duration T is heuristically chosen to be proportional to the distance between two vertices, as in the equation

$$T = \frac{\|x_{end} - x_{start}\|}{l_{unit}} T_{unit}, \tag{13}$$

where l_{unit} is a predetermined unit distance that corresponds to a duration T_{unit}.

With the above information, motion primitives can be generated as polynomial trajectories with the algorithm from [7]. By utilizing differential flatness, these polynomials are used to acquire the state of the UAV at any time instance. With the use of this method it is only possible for the UAV to have an attitude where $\psi = 0$. However, the ψ is variable in the motion primitives generated by the obstacle avoidance algorithm.

B. Collision Detection

When searching for G_f it is necessary to take the shape of the UAV and the obstacles into account as well as being able to detect collision with high precision. Therefore, the obstacles and the UAV are modelled as convex polytopes in the shape of cuboids, by using half-plane representation (H-representation).

$$x \in \mathbb{R}^3 : \mathrm{Ax} \leq \mathrm{b} \tag{14}$$

The rows in matrix A contains the normal vectors of the surfaces of the polytopes and the rows of b sets the distance to the surface from its center.

With H-representation it is possible to detect collision by checking for intersection between two polytopes,

$$H_1 := \{x \in \mathbb{R}^3 : \mathrm{A}_1\mathrm{x} \leq \mathrm{b}_1\} \tag{15}$$

and

$$H_2 := \{x \in \mathbb{R}^3 : \mathrm{A}_2\mathrm{x} \leq \mathrm{b}_2\}. \tag{16}$$

If the two polytopes would intersect, there exists an x that solves both (15) and (16).

To check for collision between the UAV and obstacles while the UAV is in motion, the state of the UAV is updated by considering each of its eight physical vertices χ. Since the center of mass $x(t)$ as well as the rotation $R(t)$ is known for the UAV for every time instance throughout the trajectory, the vertices can be rotated from B to W using (17), as defined in [9].

$$\chi' = R(t)\chi + x(t) \tag{17}$$

The updated positions of the vertices χ' can then be converted back to H-representation, making it possible to check for intersection at the current state of the UAV.

The intersection checking between the polytopes as well as the handling of the polytope representations are made by the *Python* package *polytope*.

Furthermore, to decrease the computational time, collision checking is only done when the bounding spheres of the UAV and an obstacle intersects. In other words, when the distance between their respective centers is less then the sum of their respective sphere radii.

C. Obstacle Avoidance

Obstacle avoidance is a key feature in being able to find G_f in a complex environment where aggressive maneuvers are necessary. The purpose of implementing obstacle avoidance is to combine it with RRT* when a motion primitive collides with an obstacle. The algorithm should in this case attempt to safely steer past the obstacle.

As formulated in [11], obstacle avoidance can be described as remaining within a safety set $\mathcal{C}$ described by CBFs. Therefore, $\mathcal{C}$ can be defined by a continuously differentiable CBF $h : \mathbb{R}^n \to \mathbb{R}$ as

$$\mathcal{C} = \{X \in \mathbb{R}^n : h(X) \geq 0\}. \tag{18}$$

With the use of CBFs the obstacle avoidance algorithm can be explained as a safety filter, as seen in Fig. 4, that receives a control signal u_{des} and outputs a filtered control signal u^* that ensures safety for the UAV. u_{des} is generated from a controller for position based flight mode, introduced in [10], with the purpose of steering the UAV to σ_{end} regardless of its current state. This controller is explained more in depth in subsection IV-D2. The pseudo code for the obstacle avoidance is shown at the end of this subsection.

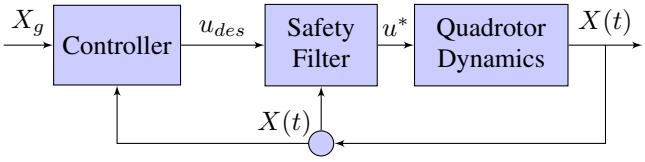


Fig. 4. Schematic of how the obstacle avoidance is implemented.

This project utilizes a variant of CBF called High-Order Control Barrier Functions (HOCBF), explained in [15], as it is consistent with the control signal u. We can now define a HOCBF of order r as a rth-order differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ together with a series of functions,

$$\Psi_0 = h(X), \ \Psi_k = (\frac{\partial}{\partial t} + \alpha_k)\Psi_{k-1}, \ 1 \leq k \leq r. \tag{19}$$

With the use of HOCBF we can formulate the following safety constraints:

$$L_{\mathfrak{f}}\Psi_{r-1}(X) + L_{\mathfrak{g}}\Psi_{r-1}(X)u + \alpha_r\Psi_{r-1}(X) \geq 0, \ X \in \mathcal{C}, \tag{20}$$

where

$$L_{\mathfrak{f}}\Psi_{r-1} = \frac{\partial\Psi_{r-1}}{\partial X}\mathfrak{f}(X), \quad L_{\mathfrak{g}}\Psi_{r-1} = \frac{\partial\Psi_{r-1}}{\partial X}\mathfrak{g}(X). \tag{21}$$

With the result from [15] these equations will guarantee safety for the UAV.

More specifically, we have chosen a 2nd order HOCBF ($r = 2$). With the use of (19) and (20), we can define the function

$$\Psi_1 = (\frac{\partial}{\partial t} + \alpha_1)\Psi_0, \tag{22}$$

together with the associated safety constraints

$$L_{\mathfrak{f}}\Psi_1(X) + L_{\mathfrak{g}}\Psi_1(X)u + \alpha_2\Psi_1(X) \geq 0, \ X \in \mathcal{C}. \tag{23}$$

The implementation of $h(X)$ is based on using sixteen evenly spread points P_i, $i = 1...16$ inside the UAV cuboid, illustrated in Fig. 5. Each point represents the center of a sphere with the radius r_s chosen to cover the whole UAV cuboid in spheres. By determining the closest point $Q_{i,j}$, which changes as the state of the UAV is updated, on the jth obstacle polytope O_j from P_i as

$$Q_{i,j} = \text{argmin } ||P_i - O_j||, \tag{24}$$

we can take the boundaries of the obstacles into account. By determining a point $P_{r_{i,j}}$, in (25), on the surface of the ith sphere in the direction towards $Q_{i,j}$ as shown in Fig. 5, we are also able to take the boundaries of the UAV into account.

$$P_{r_{i,j}} = P_i + \frac{(P_i - Q_{k,i,j})}{\|P_i - Q_{k,i,j})\|}r_s \tag{25}$$

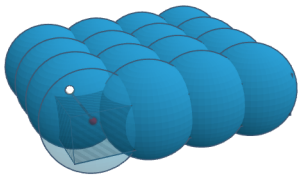


Fig. 5. Visualization of the spheres covering the UAV cuboid, P_i in red and $P_{r_{i,j}}$ in white.

With this design of $h(x)$ we are able of maintaining the shape of the UAV, so that $\min(h_{i,j}(X)) = 0$ at the instance the UAV collides with an obstacle. This allows $h(X)$ to define the unsafe set with great precision, while enabling aggressive maneuvers. With the calculated sphere surface points, $h(x)$ is determined as

$$h_{i,j}(X) = \|P_{r_{i,j}} - Q_{i,j}\|, \tag{26}$$

with the time derivative

$$\dot{h}_{i,j}(X) = \frac{((P_{r,i} - P_i)R^{-1}\dot{R} + \dot{x})}{\|P_{r_{i,j}} - Q_{i,j}\|}(P_{r,i} - Q_{k,i,j}). \tag{27}$$

Each combination of P_i and O_j corresponds to a separate instance of (23), which are used as constraints in solving the quadratic programming (QP) problem in (28). The QP finds a feasible solution u^* by minimizing the difference between the desired control signal u_{des} and the output u from the safety constraints. Additional constraints are also added to limit the control input of the UAV.

$$\begin{aligned} u^*(X) = \underset{u\in\mathbb{R}^4}{\text{argmin}} \quad & ||u - u_{des}(X,t)||^2 \\ s.t. \quad & L_{\mathfrak{f}}\Psi_1(X) + L_{\mathfrak{g}}\Psi_1(X)u \geq -\alpha_2\Psi_1(X) \end{aligned} \tag{28}$$

The obstacle avoidance algorithm generates a sampled motion primitive with a certain time interval Δt. For each sample a new u_{des} as well as the CBFs for every P_i are calculated. Because of this, a new state $X(t)$ is determined at every sample. To determine the next state $X(t+\Delta t)$, the latest computed output u^* together with the latest state $X(t)$ is used in (7) so that $X(t+\Delta t)$ is determined as the integration of $\dot{X}$ added to $X(t)$. In order to avoid collision throughout the whole motion primitive, $h_{i,j} > 0$ needs to be true at every sample. If the QP solver cannot find a solution that maintains u^* within the constraints, the motion primitive is deemed infeasible.

Algorithm 2 Obstacle avoidance

Input: q_{start}, q_{end} and α_1, α_2
Output: σ_f between q_{start} and q_{end}

$state = q_{start}.state$
while $state != q_{end}.state$ **do**
  $u_{des} \leftarrow get_u_{des}(state)$
  $h, \dot{h} \leftarrow compute_h(state)$
  $\Psi_1 = \dot{h} + \alpha_1 h$
  $\mathfrak{f} \leftarrow get_f(state)$
  $\mathfrak{g} \leftarrow get_g(state)$
  $L_\mathfrak{f}\Psi_1 \leftarrow get_L_\mathfrak{f}\Psi_1(\mathfrak{f}, \Psi_1)$
  $L_\mathfrak{g}\Psi_1 \leftarrow get_L_\mathfrak{g}\Psi_1(\mathfrak{g}, \Psi_1)$
  if $L_\mathfrak{f}\Psi_1 + L_\mathfrak{g}\Psi_1 u_{des} + \alpha_2\Psi_1 < 0$ **then**
    try:
      $u^* \leftarrow solve_QP(u_{des}, L_\mathfrak{f}\Psi_1, L_\mathfrak{g}\Psi_1, \alpha_2\Psi_1)$
    except:
      break
  else
    $u^* = u_{des}$
  end if
  $state \leftarrow update_state(state, u^*)$
  $\sigma_f.append(state)$
end while
return σ_f

D. Controllers

In this project there are two different types of controllers. One of them is utilized to track the motion planner computed trajectory G_f. This controller is a velocity and attitude based flight mode controller implemented by [9], without any modifications. The second one is used in the obstacle avoidance algorithm. This controller is a position based flight mode controller, implemented to generate u_{des} to steer the UAV in the direction towards σ_{end} at every sample regardless of the current state of the UAV. Both controllers are implementations based on the work of [10].

1) Attitude and Velocity Controller: The attitude part of the controller is used to adjust the moment M based on a desired attitude R_d, while the velocity part of the controller use the desired velocity v_d to determine the thrust magnitude, as seen in Fig. 6. From the velocity, attitude and angular velocity tracking errors, defined as

$$e_v = v - v_d \tag{29}$$

$$e_R = \frac{1}{2}\left(R_d^T R - R^T R_d\right)^\vee \tag{30}$$

$$e_\Omega = \Omega - R^T R_d \Omega_d, \tag{31}$$

the moment and the thrust is calculated as in [10, Eq. 11 and Eq. 38]. The operator $^\vee$: $\mathbb{R}^{3\times3} \rightarrow \mathbb{R}^3$ is the inverse hat operator $[.]_\times$.

2) Position Controller: The position based flight mode controller used in the obstacle avoidance algorithm, as seen in Fig. 4, takes a desired position x_d (chosen as x_{end} in this paper), a control attitude R_c and a control angular velocity $\hat{\Omega}_c$, further defined in [10]. Combined with the tracking errors for the position and velocity together with the error in attitude and velocity defined as

$$e_x = x - x_d \tag{32}$$

$$e_v = v - \dot{x}_d \tag{33}$$

$$e_R = \frac{1}{2}\left(R_c^T R - R^T R_c\right)^\vee \tag{34}$$

$$e_\Omega = \Omega - R^T R_c \Omega_c, \tag{35}$$

the desired control signal u_{des} is calculated with (36) and (37). Since x_d is stationary, the magnitude of both $\dot{x}_d$ and $\ddot{x}_d$ are zero. All constants k_x, k_v, k_R and k_Ω are positive.

$$f = (m\ddot{x}_d - mgz_W - k_x e_x - k_v e_v) \cdot Rz_W \tag{36}$$

$$\begin{aligned} M = &- k_R e_R - k_\Omega e_\Omega + \Omega \times J\Omega \\ &- J\left(\hat{\Omega} R^T R_c \Omega_c - R^T R_c \dot{\Omega}_c\right) \end{aligned} \tag{37}$$

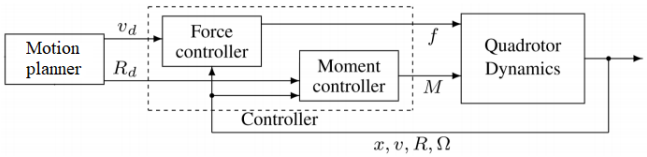


Fig. 6. Illustration of the closed-loop system with the controller in IV-D1

V. Results

The motion planning algorithm was implemented in *Python 3.7* and is available on *Github* [16]. Visualisations of the computed trajectory from the motion planner algorithm were done in *Matlab* and for the simulations to track the computed trajectory, the closed-loop system in Fig. 6 was used in *Simulink*. Gaussian white noise was also added to the control signal to represent disturbances in the actuators of the UAV, with a signal to noise ratio of $10\,\mathrm{dB}$. The physical parameters of the UAV was chosen to be the following:

- total mass of $m = 1\,\mathrm{kg}$.
- inertial matrix $J = \begin{pmatrix} 0.082 & 0 & 0 \\ 0 & 0.0845 & 0 \\ 0 & 0 & 0.1377 \end{pmatrix} \mathrm{kgm}^2$
- width of $0.5\,\mathrm{m}$ and a height of $0.1\,\mathrm{m}$.

The constants for the attitude and velocity controller were set to $k_v = 40$, $k_R = 50$ and $k_\Omega = 2.5$. The constants for the position controller in the obstacle avoidance were set to $k_x = 7$, $k_v = 8$, $k_R = 15$ and $k_\Omega = 2.6$.

To demonstrate the results of this project three different scenarios were tested:

1) RRT* (without obstacle avoidance sub-module) in a cube obstacle cluttered environment.
2) Obstacle avoidance algorithm, testing aggressive maneuvers through a narrow gap.
3) The complete motion planner, RRT* combined with obstacle avoidance, in a more complex environment with narrow vertical gaps and horizontal barriers.

Each scenario had a set space interval for each axis i in x_{rand} as $0 \leq x_{i,rand} \leq 5$ and for the angles a_i for each axis in R_{rand} as $|a_i| \leq 1.3\,\mathrm{rad}$. To bound the trajectory six obstacles forming a box were added, that are not shown for illustrative purposes.

In RRT*, collision checking was done at $50\,\mathrm{Hz}$ to make sure the trajectory was collision free and the obstacle avoidance was sampled at $250\,\mathrm{Hz}$, to make sure the UAV was in the safe set. The CBF constants were tuned to $\alpha_1 = 20$ and $\alpha_2 = 70$ and the actuation limit of u^* in the QP was set to

$$u^* = \begin{pmatrix} f \\ M_1 \\ M_2 \\ M_3 \end{pmatrix} \leq \begin{pmatrix} 40 \\ 40 \\ 40 \\ 40 \end{pmatrix}. \tag{38}$$

A. Scenario 1: RRT*

This scenario demonstrated the ability of RRT* to find an optimal trajectory from an initial state to a goal state in an environment cluttered with cube obstacles as shown in Fig. 8. The number of randomly generated vertices was set to $K = 350$ and the incremental distance $\Delta q = 3\,\mathrm{m}$ was used. The initial state was set to $X_0 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, vec(R_0)^T]$ and the goal state was set to $X_g = [4, 4, 4, 0, 0, 0, 0, 0, 0, 0, vec(R_g)^T]$, where $R_0 = R_g = I$.

The resulting trajectory consisted of three motion primitives and four vertices and the total duration was $4.24\,\mathrm{s}$. As visualised in Fig. 8 with the computed trajectory from the motion planner, the UAV avoided collisions throughout the whole trajectory and reached X_g. This was also true for the simulation of the trajectory in Fig. 9, which was shown to be almost the same as the computed trajectory with a minor difference in tilt at the end of the trajectories.

For the numerical trails, X_g was reached after 56 randomly generated vertices at the total cost in jerk of $23\,510\,\mathrm{m}^2/\mathrm{s}^6$. After 350 randomly generated vertices the cost was decreased to $634\,\mathrm{m}^2/\mathrm{s}^6$.

The execution time and the number of randomly generated vertices that was required for RRT* to find a feasible path was measured ten times for ten different Δq. The average execution time and the average number of generated vertices are shown in Fig. 7 with the lowest average execution time of $11.1\,\mathrm{s}$ when $\Delta q = 3\,\mathrm{m}$.

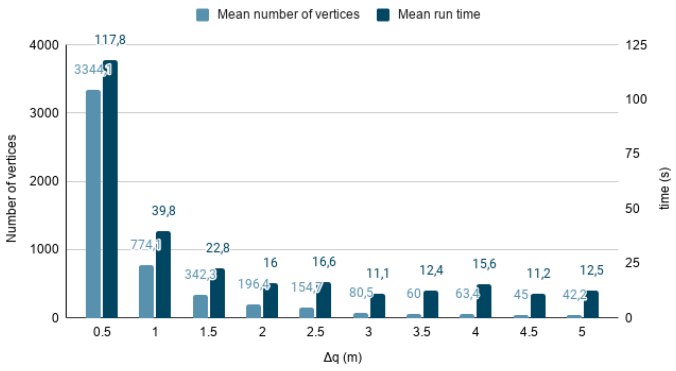


Fig. 7. Scenario 1: Diagram of mean results from 100 runs.

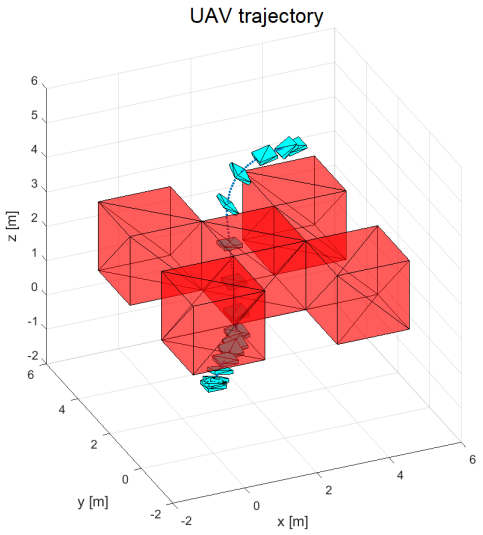


Fig. 8. Scenario 1: Computed trajectory with RRT*, $K = 350$, $\Delta q = 3\,\mathrm{m}$.

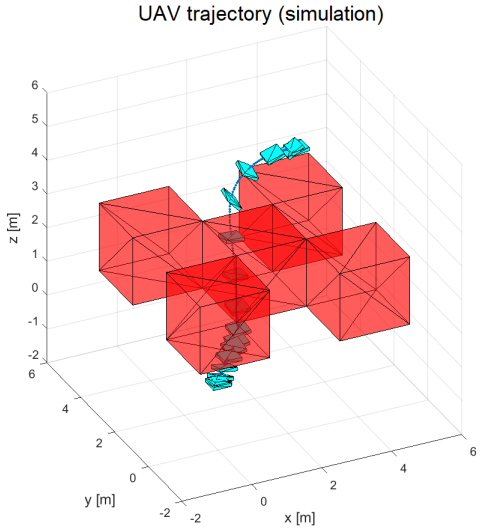


Fig. 9. Scenario 1: Simulated trajectory with RRT*, $K = 350$, $\Delta q = 3\,\mathrm{m}$.

B. Scenario 2: Obstacle Avoidance

In order to test the capability of the obstacle avoidance as well as aggressive maneuvering, the algorithm was set to navigate through a narrow gap in a wall, as shown in Fig. 10. The initial state was set to $X_0 = [1.615, 2.18, 1, 0, 0, 0, 0, 0, 0, 0, vec(R_0)^T]$ and the goal state was set to $X_g = [4.5, 2.86, 2.2, 0, 0, 0, 0, 0, 0, 0, vec(R_g)^T]$, where $R_0 = R_g = I$.

The resulting computed trajectory in Fig. 10 was collision free and reached X_g for both the trajectory computed by the motion planner and for the simulation of the trajectory in Fig. 11. Furthermore, it is confirmed from the simulation that the UAV was able to manage aggressive maneuvers when using the obstacle avoidance, even when exposed to disturbances in the actuators, since the trajectory in Fig. 10 and Fig. 11 was almost exactly the same, with minor differences near the goal of the trajectory.

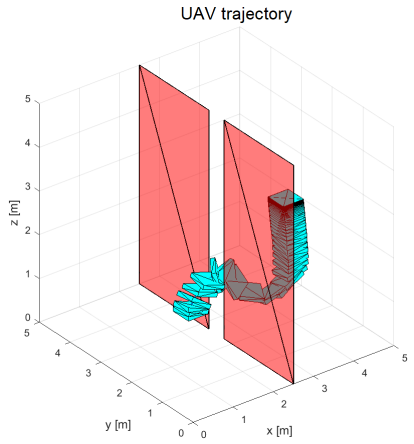


Fig. 10. Scenario 2: Computed trajectory with obstacle avoidance algorithm, $\alpha_1 = 20$, $\alpha_2 = 70$.

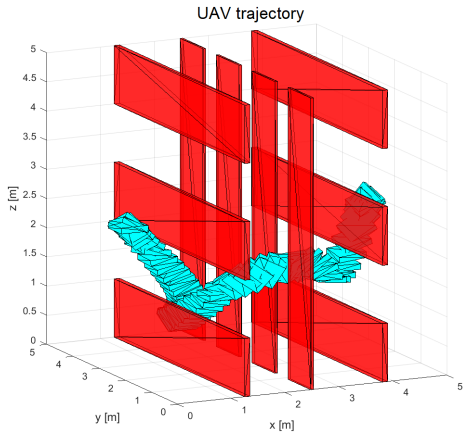


Fig. 12. Scenario 3: Computed trajectory with the complete motion planner.

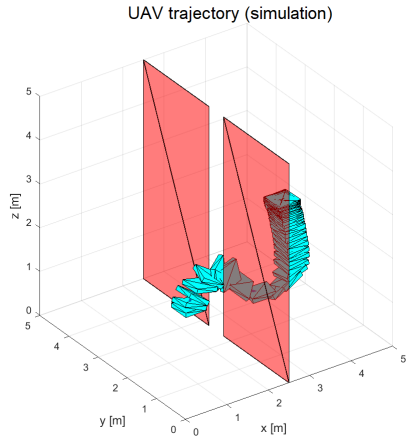


Fig. 11. Scenario 2: Simulated trajectory with obstacle avoidance algorithm, $\alpha_1 = 20$, $\alpha_2 = 70$.

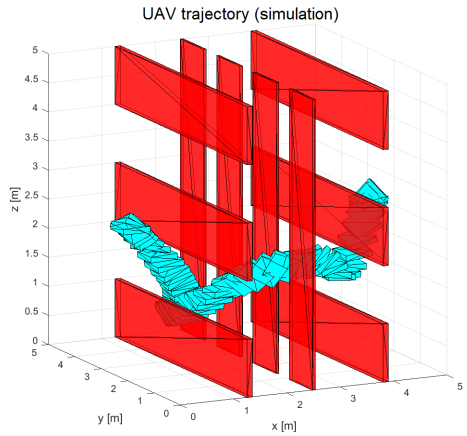


Fig. 13. Scenario 3: Simulated trajectory with the complete motion planner.

C. Scenario 3: Complete Motion Planner

In this scenario, RRT* was combined with the obstacle avoidance algorithm for the purpose of allowing the complete motion planner to manage complex environments. To model this environment, two rows of barriers, both made out of three stacked thin obstacles, were placed on each side of the enclosed space and between them a thin wall with three vertical narrow gaps. These obstacles are visualised in Fig. 12. The number of randomly generated vertices was set to $K = 2500$ with $\Delta q = 1.5\,\mathrm{m}$. The initial state was set to $X_0 = [0.3, 2.5, 2.5, 0, 0, 0, 0, 0, 0, vec(R_0)^T]$ and the goal state was set to $X_g = [4.7, 2.5, 2.5, 0, 0, 0, 0, 0, 0, vec(R_g)^T]$ where $R_0 = R_g = I$.

The total duration of the high-level trajectory was $4.38\,\mathrm{s}$, consisting of five motion primitives and six vertices. As shown in Fig. 12 the computed trajectory from the motion planner was collision free and reached the high-level goal, which was also true for the simulated trajectory in Fig. 13. In the beginning of the trajectory, after the first barrier of obstacles, it is seen that it made a slight detour.

The high-level trajectory was found after 118 randomly generated vertices at the total cost in jerk of $143.6 \times 10^6\,\mathrm{m}^2/\mathrm{s}^6$. This was reduced after 2041 generated vertices to $6750\,\mathrm{m}^2/\mathrm{s}^6$.

VI. Discussion

The three different simulation scenarios proves the competence of the resulting motion planner developed in this project. The usage of RRT* enables the motion planner to find a feasible high-level trajectory in a various range of start and end states. The great cost reduction that occurred after the high-level goal was found demonstrates that the rewiring in RRT* works successfully to minimize the total cost of the trajectory.

In scenario 1 it was shown that the average execution time required to find X_g was the shortest when $\Delta q = 3\,\mathrm{m}$. This result is probably dependent on the environment in which the motion planner is used in, implying that the most effective choice of Δq will differ between different scenarios. Overall, RRT* is capable of finding a feasible path if there are no narrow spaces that requires aggressive maneuvers.

In scenario 2 the obstacle avoidance algorithm was shown to keep the UAV in the safe set, but with a low margin to collision, making it necessary allowing a high magnitude of u^* and therefore a high cost in jerk. So, the later the obstacle avoidance algorithm reacts to an obstacle, the greater the magnitude of u^* will need to be to find a feasible trajectory past the obstacle. This poses a challenging problem when the

QP considers the constraints of the UAV actuators, making it difficult for the obstacle avoidance algorithm to find a feasible solution. Therefore, feasibility is gained when generating a trajectory from only certain angles, resulting in the need for many attempts from different angles, making it great combining with RRT*. However, this combination requires a long computational time, since just one attempt alone requires a relatively long computational time as a high sampling rate is necessary.

Another common issue with CBFs is that the state of the UAV can get stuck in local equilibrium. This depends partially on the type of controller that generates u_{des}. Since a position controller is used in this paper it is highly possible that it amplifies this issue, seeing that this type of controller only considers the goal position and will therefore always steer the UAV in the direction towards the goal.

The complete motion planner with the combination of RRT* and obstacle avoidance in scenario 3 demonstrates the potential of this method. The complete motion planner finds a path in an environment with sequentially small clearance in both x and y direction. This forces the motion planner to attempt obstacle avoidance at multiple occasions in order to find a feasible path. In the trajectory generated in scenario 3 the obstacle avoidance algorithm is seen to navigate the UAV through a narrow gap in a smoother motion than it did in scenario 2. This is due to the rewiring minimizing its cost, which was made possible by dividing the motion primitive generated by the obstacle avoidance algorithm into several smaller motion primitives. This increased the effects of the rewiring, which can be seen from the great cost reduction that occurred in scenario 3. There is also a detour right after the first barrier of obstacles in scenario 3, which implicates that it possible to optimize this trajectory much more. Therefore, the detour will most likely be evened out if the rewiring would proceed with more iterations. When the complete motion planner found a feasible high-level trajectory, it resumed with the rewiring to optimize the path. As this is done without the obstacle avoidance algorithm, the computational time for every iteration in RRT* was drastically decreased.

As the motion primitive generator from [14] does not account for ψ, as opposed to the obstacle avoidance algorithm, it is important to take that into account at the end of a motion primitive generated by the obstacle avoidance algorithm. This is accomplished with the position controller, that uses the x axis of B in X_g to determine the control attitude, where $\psi = 0$ if the next motion primitive was generated by the algorithm in [7].

A. Method Limitations

The method presented has some limitations regarding real-world applications. Firstly, it is based on the assumption that the obstacles can be defined mathematically, which is usually not the case in a real-world situation. Secondly, the obstacles are modelled as cuboids making it easier to determine $Q_{i,j}$ in subsection IV-C. Thirdly, the obstacle avoidance often require a too high control signal, which in some cases cannot be reduced by the rewiring in RRT*. Lastly, the motion planner requires a long computational time so it can only serve as an offline planner.

B. Potential Improvements

To begin with, a future improvement would be to take the constraints of an actual quadrotor into account when generating the trajectory, in other words making it possible to manage aggressive maneuvers with more strict constraints. Too reduce the long computational time for the motion planner algorithm, the implementation in code could be done with *C++* instead of *Python*. Furthermore, one could upgrade the obstacle avoidance algorithm by making it more computationally efficient, through e.g. decreasing the number of $h(x)$ functions needed to make the obstacle avoidance work properly. The robustness of the obstacle avoidance algorithm could also be improved, by making it able to react earlier and in that way reduce the need for high control signals. Another possible improvement would be to make the motion planner handle more general obstacles.

VII. Conclusion

The result of this project is deemed successful as the resulting motion planner has in fact demonstrated to be able of managing general scenarios in a complex environment, where aggressive manoeuvres are necessary, as well as being able of effectively optimizing the trajectory. This was also a success when using the controller in Fig. 6 to track the computed trajectory with added disturbances to the control signal. These results have been accomplished with the use of aggressive maneuvers, consideration of the UAV shape and by minimizing a cost in jerk of the high-level trajectory.

Although the main goals of the project were reached, there are still several aspects in which the motion planner could be improved. Improvements concerning the computational time would be a high priority to make this algorithm work in a real-world scenario. Other improvements would be to optimize the obstacle avoidance algorithm in several ways, e.g. increase the robustness and make the algorithm work for stricter constraints.

Acknowledgment

The authors would like to thank Xiao Tan for his time and commitment to continuously support this project with useful insights of previous research on this subject.

References

[1] Y. Bestaoui Sebbane, *"Smart autonomous aircraft: flight control and planning for UAV"*, 1st ed., ser. Chapman & Hall Book. Boca Raton: O'Reilly Media, Oct. 2015.

[2] C. Luo, W. Miao, H. Ullah, S. McClean, G. Parr, and G. Min, *"Unmanned Aerial Vehicles for Disaster Management"*. Singapore: Springer, Aug. 2019, pp. 83–107.

[3] Z. Zaheer, A. Usmani, E. Khan, and M. A. Qadeer, "Aerial surveillance system using UAV," in *2016 Thirteenth International Conference on Wireless and Optical Communications Networks (WOCN)*, Jul. 2016, pp. 1–7.

[4] (2021, Apr.) Unmanned aerial vehicle (UAV). Techopedia, Edmonton, Alberta, USA. [Online]. Available: https://www.techopedia.com/definition/29896/unmanned-aerial-vehicle-uav

[5] V. Becerra, *Autonomous Control of Unmanned Aerial Vehicles*. Basel: Switzerland: AMDPI - Multidisciplinary Digital Publishing Institute, Jun. 2019.
[6] S. Aggarwal and N. Kumar, "Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges," *Computer Communications*, vol. 149, pp. 270–299, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366419308539
[7] M. W. Mueller, M. Hehn, and R. D'Andrea, "A computationally efficient motion primitive for quadrocopter trajectory generation," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1294–1310, Dec. 2015.
[8] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 2520–2525.
[9] O. Palfelt and F. Skjernov, "Motion planning for aggressive flights of an unmanned aerial vehicle," BsC thesis, KTH, Sweden, Stockholm, May 2020.
[10] T. Lee, M. Leok, and N. Mcclamroch, "Control of complex maneuvers for a quadrotor UAV using geometric methods on SE(3)," vol. 4, Mar. 2010. [Online]. Available: https://arxiv.org/abs/1003.2005
[11] G. Yang, B. Vang, Z. Serlin, C. Belta, and R. Tron, "Sampling-based motion planning via control barrier functions," *Proceedings of the 2019 3rd International Conference on Automation, Control and Robots*, Oct. 2019. [Online]. Available: http://dx.doi.org/10.1145/3365265.3365282
[12] S. LaValle, "Rapidly-exploring random trees: a new tool for path planning," *The Annual Research Report*, Jun. 1998. [Online]. Available: https://ci.nii.ac.jp/naid/10014962955/en/
[13] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *CoRR*, vol. abs/1105.1186, May 2011. [Online]. Available: http://arxiv.org/abs/1105.1186
[14] M. W. Meuller. (2021, Apr.) Quadrocopter trajectory generator. GitHub, San Franciso, CA. [Online]. Available: https://github.com/markwmuller/RapidQuadrocopterTrajectories
[15] X. Tan, W. S. Cortez, and D. V. Dimarogonas, "High-order barrier functions: Robustness, safety and performance-critical control," 2021, under review. [Online]. Available: https://arxiv.org/abs/2104.00101
[16] A. Medén and E. Warberg. (2021, May) Kex2021. GitHub, San Franciso, CA. [Online]. Available: https://gits-15.sys.kth.se/warberg/KEX2021

Cooperative Control of Autonomous Ground Vehicles

Mohammed Akif and Sebastian Geivald

***Abstract*—As autonomous ground vehicles grow in popularity, it is of interest to study how they could coordinate together and how the technical systems can be implemented in a safe and effective manner. The objective of this report is to examine how to autonomously move a formation of vehicles without collisions with obstacles or other vehicles. This is done by considering three fundamental aspects: trajectory tracking, formation control and collision avoidance. Firstly a trajectory tracking controller for an individual vehicle is implemented, with the function of following a desired trajectory. Secondly a displacement-based formation control is explored for two models, the double-integrator model and the nonholonomic model, with the objective of coordinating multiple vehicles to keep a certain formation. Lastly collision avoidance is integrated in the formation control by adding a repulsive term to the formation controller. It is shown that the agents maintained formation while avoiding collision with obstacles and other agents. The implemented controllers were verified through simulations in MATLAB.**

***Sammanfattning*—Eftersom autonoma markfordon blir allt mer vanligt är det av vikt att studera hur de kan samordna tillsammans och hur de tekniska systemen kan implementeras på ett säkert samt effektivt sätt. Syftet med denna rapport är att undersöka hur man autonomt kan flytta en formation av fordon utan kollisioner med hinder eller med andra fordon. Detta görs genom att tre grundläggande aspekter övervägs: projektilspårning, formationshållning och kollisionsundvikande. Först implementeras en regulator för projektilsspårning, där funktionen är att följa en önskad bana. Därefter undersöks två modeller inom förskjutningsbaserad formationshållning, med ambitionen att samordna alla fordon för att behålla formationen. Slutligen så integreras metoder för kollisionsundvikning med formationshållning genom att lägga till bortstötande teknik i regulatorn för formationshållning. Det visades att fordonen lyckades med att upprätthålla formationen samtidigt som kollisioner mellan hinder och andra fordon undveks. De implementerade regulatorerna verifierades genom simuleringar i MATLAB.**

***Index Terms*—Trajectory tracking, Formation control, Collision avoidance, multi-agent system, Autonomous ground vehicles.**

Supervisor: Fei Chen

TRITA number: TRITA-EECS-EX-2021:142

I. Introduction

The research around autonomous systems are rapidly growing [1]. As the efficiency of automation is undeniable and more repetitive tasks are successfully automated, the increasing interest of autonomous ground vehicles (AGVs) is not surprising. The possibilities of AGVs are clear and very compelling [2]. Especially usage within autonomous driving, robot exploration and rescue to name a few. In the subject of autonomous driving, replacing a human driver has several efficiency benefits. In particular regarding safety, Eco-driving and prevention of traffic jams. As for robot exploration and rescue it is about enabling the possibility of reaching areas a human can not.

Coordinating multiple AGVs to achieve a common goal is of interest in a magnitude of areas. As the need of a multi-agent system is apparent so is the research within the field [3]. The objective of this project is to explore ways of safely coordinating numerous AGVs to move within space constraints, such as to a specified location without colliding with each other or other obstacles. To achieve this, three main areas are being investigated: trajectory tracking, formation control and collision avoidance. All three research areas are fundamental concepts within control theory of AGVs. Trajectory tracking is the first cornerstone crucially needed for an AGV to follow a desired path. The two other concepts require this to work, because without a working trajectory tracking the vehicles can not be controlled. The second studied concept is formation control, essential to coordinate multiple agents simultaneously. With the help of local sensors on each agent and a centralized communication network, a desired formation can be achieved. Meaning the AGVs are coordinating themselves to form a certain structure. The reason for a formation is to create an organized structure to avoid colliding with each other, while maintaining a relatively close distance so local communication systems are in range. The third and last concept investigated in this project is collision avoidance. This includes not colliding with static obstacles while maintaining the formation in a relative sense, thus if an obstacle is in the way, all agents within the formation need to coordinate to avoid colliding with the obstacle and each other while maintaining the formation.

The report structure is as follows: section II presents the preliminaries, section III describes the problem formulation, section IV explains the differential drive model, section V-VII goes in more detail for each research area, section VIII presents and explains simulations while the results of these are discussed in section IX and finally concluded in section X.

II. Preliminaries

A. Notations

In this section, mathematical notations used in this paper are presented. A capital letter written in bold font denotes a matrix e.g. $\boldsymbol{A}$. A vector is written as a small bold letter e.g. $\boldsymbol{v}$. Given a matrix $\boldsymbol{A}$, $\boldsymbol{A}^T$ denotes the transpose matrix of $\boldsymbol{A}$. The Newton's notation (dot notation) is used for time derivatives e.g. $\dot{y}$ is the first derivative of y with respect to t. N_i is a set of agent neighbors.

B. Theory

1) Chained form: Let $x_1, x_2, ..., x_n$ be the state variables and u_1, u_2 the control variables of a system. According to [3] the chained form can be defined by:

$$\dot{x}_1 = u_1, \dot{x}_2 = u_1 x_3, ..., \dot{x}_{n-1} = u_1 x_n, \dot{x}_n = u_2 \quad (1)$$

2) Forward Euler method: Forward Euler method is an explicit numerical method for solving ordinary differential equations (ODEs) with a given initial value. An ODE written on the following form $\dot{u}(t) = f(t, u(t))$ with initial value $u(t_0) = u_0$ can be solved by $u_n = u_{n-1} + hf(t_{n-1}, u_{n-1})$ where h is the time step, $t_n = t_{n-1} + h$ and $i = 1, 2, 3, ...$

III. Problem Description

A communication network that uses closed-loop control of systems can be used for a team of autonomous ground vehicles to achieve tasks such as trajectory tracking, keeping a certain formation or avoiding collisions. These three aspects are being studied in this paper.

A. Trajectory Tracking

Trajectory tracking is a key element that has to be achieved in order to accomplish autonomous driving. The main goal of it is for a team of AGVs to follow a predefined (desired) trajectory. Motion constraints like nonholonomic constraints as well as boundary conditions need to be satisfied. This is achieved by designing a trajectory tracking controller so that the position error converges to zero asymptotically.

B. Formation Control

To achieve a team task formation control is beneficial. It is used to maintain a predetermined geometric pattern when a multi-agent system is moving. In this paper displacement-based formation control is studied. Each agent actively controls its neighbours displacement in order to achieve a desired formation. Displacement-based formation in two different cases is considered in this project; the double-integrator modeled agent case and the nonholonomic agent model case.

C. Collision Avoidance

Collision avoidance is a necessary feature for AGVs to avoid collisions between agents and other obstacles while keeping a certain formation. Potential fields are used in this project. A repulsive force is generated when an agent senses an obstacle or another agent is located at a shorter distance than a predefined minimum distance.

IV. Differential Drive Vehicle

An autonomous ground vehicle can be represented by the differential drive vehicle model in [3], seen in figure 1. It is assumed wheels on each side of the vehicle have the same angular velocity and size. The coordinates of the vehicle's center are defined by the vector $\left[x, y, \theta\right]^T$, where x and y describe the position of the vehicle on the XY-plane and θ is the heading angle of the vehicle.

It is assumed that there are no side slipping when the vehicle is rolling, meaning there is no movement along the direction of the wheel axles. This is described by the following nonholonomic constraint $\dot{x}\sin\theta - \dot{y}\cos\theta = 0$. Nonholonomic means that the state of the system is only depending on the path taken by the vehicle.

The differential-drive vehicle has the following kinematic model, from [3]:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}\cos\theta\\ \sin\theta\\ 0\end{bmatrix} u_1 + \begin{bmatrix}0\\ 0\\ 1\end{bmatrix} u_2 \quad (2)$$

where u_1 and u_2 are the kinematic control variables corresponding to the driving velocity and the steering velocity, respectively.

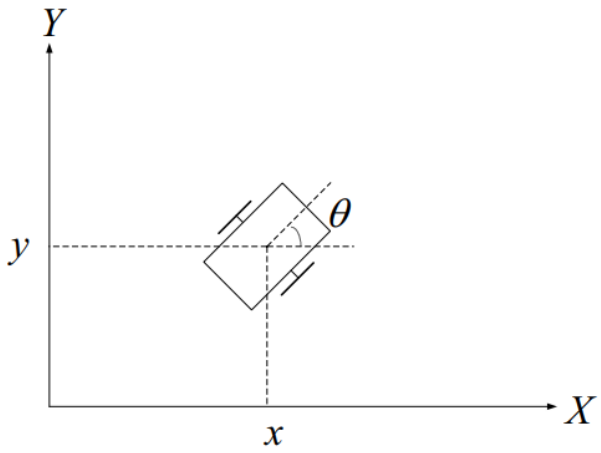


Fig. 1. A differential-drive vehicle model, from [3]

V. Trajectory Tracking

A. Objective

The purpose of this section is to design a feedback controller, namely a tracking controller which ensures that the differential-drive vehicle in model (2) is following the desired trajectory asymptotically.

B. Tracking Control Design

1) Conversion to chained form: It is easier to design the tracking controller when working with linear equations instead of nonlinear equations. Thus the nonlinear system (2) is transformed into the chained form. This is done in three steps; transforming the system's coordinates, mapping the control variables and using equation (1).

The coordinate transformation and control mapping of the kinematic model are given by [3]:

$$x_1 = \theta, \quad x_2 = x\sin\theta - y\cos\theta, \quad x_3 = x\cos\theta + y\sin\theta \quad (3)$$

$$u_1 = v_2 + x_3 v_1, \quad u_2 = v_1 \quad (4)$$

Equation (1) now gives the kinematic model (2) in chained form:

$$\dot{x}_1 = v_1, \quad \dot{x}_2 = v_1 x_3, \quad \dot{x}_3 = v_2 \quad (5)$$

2) Error system: An error system is introduced to verify that the state tracking error converges asymptotically to zero. The state tacking error is defined as the distance between the present position and the desired position:

$$\boldsymbol{x}_e = \begin{bmatrix} x_{1e}, x_{2e}, x_{3e} \end{bmatrix}^T \triangleq \boldsymbol{x} - \boldsymbol{x}_d \tag{6}$$

and the feedback control to be designed is defined as:

$$\boldsymbol{v} = \begin{bmatrix} v_1, v_2 \end{bmatrix}^T \triangleq \boldsymbol{u} - \boldsymbol{u}_d \tag{7}$$

3) Cascade system: Both the kinematic model $\boldsymbol{x}$ and the desired trajectory $\boldsymbol{x}_d$ are in chained form, thus the error system (6) can be described by the following cascade system from [3]:

$$\dot{x}_{1e} = v_1 \tag{8}$$

$$\dot{\boldsymbol{z}} = u_{1d}(t)\boldsymbol{A}_c\boldsymbol{z} + \boldsymbol{B}_c v_2 + \boldsymbol{G}(\boldsymbol{x}_d, \boldsymbol{z})v_1 \tag{9}$$

where $\boldsymbol{z} = \begin{bmatrix} z_1, z_2 \end{bmatrix}^T \triangleq \begin{bmatrix} x_{2e}, x_{3e} \end{bmatrix}^T$ and $\boldsymbol{G}(\boldsymbol{x}_d, \boldsymbol{z}) = \begin{bmatrix} z_2 + x_{3d}, 0 \end{bmatrix}$. Equation (9) is in the controllable form if $\boldsymbol{A}_c$ and $\boldsymbol{B}_c$ are given by $\boldsymbol{A}_c = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ and $\boldsymbol{B}_c = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$. By designing a stabilizing control v_1 the last term in equation (9) will vanish, hence it can be eliminated.

4) Control design: The tracking control design chosen by [3] is:

$$v_1 = -r_1^{-1} p_1 x_{1e}, \quad v_2 = -r_2^{-1}\boldsymbol{B}_c^T\boldsymbol{P}_2(t)\boldsymbol{z} \tag{10}$$

where $p_1 = \sqrt{q_1 r_1}$ and r_1, r_2, q_1 are positive scalar constants. $\boldsymbol{P_2}(t)$ is given by the solution to the differential Riccati equation

$$0 = \dot{\boldsymbol{P}}_2 + \boldsymbol{P}_2\boldsymbol{A}_c u_{1d}(t) + u_{1d}(t)\boldsymbol{A}_c^T\boldsymbol{P}_2 - \frac{1}{r_2}\boldsymbol{P}_2\boldsymbol{B}_c\boldsymbol{B}_c^T\boldsymbol{P}_2 + \boldsymbol{Q}_2 \tag{11}$$

where $\boldsymbol{Q}$ is a positive definite matrix.

VI. Formation Control

A. Objective

The goal in this section is to propose a controller that handles the formation control. A controller that causes the agents to form a shape of a parallelogram, meaning a quadrangle with equal opposite sides. Squares, rectangles and rhombuses are all parallelograms. Local communication instruments on each vehicle could be used to exchange data, for instance positional information. But local instruments demands to be within a specific range to operate. This requires the vehicles to maintain a close enough distance between each other. That is part of the benefit of having a formation controller coordinating the agents to an organized structure.

B. Displacement-based Formation Control

There are multiple types of formation control concepts, each with their upsides and downsides. Position-based formation has more advanced sensing capabilities while distance-based formation has more interactions between vehicles. In this project the displacement-based formation control is investigated, which is a hybrid between the two. The agents have access to both local and global communication. Global communication refers to each agent being connected to a network with a centralized system that has access to all information. This system can receive, calculate and send back data. A global network demands an extremely consistent connectivity with low delay to function reliably. If the global communication fails, the local communication could compensate. Local communication refers to each vehicle having local sensors and can gather information from their neighbours. For instance, this could mean not all agents in a formation could communicate depending on the size and shape of the formation, but the formation could still be achieved. In this project, the agents are able to sense relative positions and velocities of their neighbours with respect to a global coordinate system [4].

1) Double-integrator model: The double-integrator model is a variation of displacement-based formation control. It can be derived from the differential drive model given in (2). From [4] and [5] it is presented that a double integrator model is defined as following

$$\begin{cases} \dot{\boldsymbol{p}}_i = \boldsymbol{v}_i \\ \boldsymbol{v}_i = \boldsymbol{u}_i \end{cases} \tag{12}$$

for $i = 1, 2, ..., N$ where $\boldsymbol{p}_i \in \mathrm{R}^n$ denotes position, $\boldsymbol{v}_i \in \mathrm{R}^n$, denotes velocity and $\boldsymbol{u}_i \in \mathrm{R}^n$ denotes control input of agent i with respect to a global coordinate system. In order for the agents to achieve the target formation, the current relative position between two agents needs to coincide with the desired relative position between the same agents. This is represented as

$$\begin{aligned} \boldsymbol{E}_{p^*,v^*} := ([\boldsymbol{p}^T\boldsymbol{v}^T]^T : p_j - p_i = {p_j}^* - {p_i}^*, \\ v_j - v_i = {v_j}^* - {v_i}^*, i, j \in \nu), \end{aligned} \tag{13}$$

where j and i denote agents that are neighbours. Additionally p and $p*$ denote current and desired position. The control law can be designed as following

$$\begin{aligned} \boldsymbol{u}_i = -k[&\sum_{j \in N_i} w_{ij}(p_i - p_j - {p_i}^* + {p_j}^*) \\ &+ \sum_{j \in N_i} w_{ij}(v_i - v_j + {v_i}^* - {v_j}^*], \end{aligned} \tag{14}$$

where $k > 0$. The position summation can be replaced with $\boldsymbol{L}\boldsymbol{p}(t)$ and the velocity summation can be replaced with $\boldsymbol{L}\dot{\boldsymbol{p}}(t)$, where $\boldsymbol{L}$ is the laplacian matrix. The error dynamics follow as:

$$\begin{bmatrix} \dot{\boldsymbol{e}}_p(t) \\ \dot{\boldsymbol{e}}_v(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I}_n \\ -k\boldsymbol{L} & -k\boldsymbol{L} \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_p(t) \\ \boldsymbol{e}_v(t) \end{bmatrix}, \tag{15}$$

where the relative positions and velocity only converge to the desired ones if and only if the matrix in (15) has n zero eigenvalues while the real parts of the other eigenvalues are negative.

2) Nonholonomic model: Another variation of the displacement-based formation control is the nonholonomic agent model. The following ODE system is the kinematic model (2) with $u_1 = v_i$ and $u_2 = w_i$, given as

$$\begin{cases} \dot{x_i} = v_i \cos\theta \\ \dot{y_i} = v_i \sin\theta \\ \dot{\theta} = \quad w_i \end{cases} \tag{16}$$

for $i = 1, 2, ..., N$, where $\boldsymbol{p}_i = \begin{bmatrix} x_i & y_i \end{bmatrix}^T \in \mathbb{R}^2$ and $\theta_i \in (-\pi, \pi]$ describes the position and direction angle for agent i,

see (2). The control law for the nonholonomic model yields the derived control inputs v_i and w_i which are given by

$$\begin{aligned} v_i &= k \begin{bmatrix} \cos\theta_i & \sin\theta_i \end{bmatrix} \sum_{j \in N_i} (p_j - p_i - {p_j}^* + {p_i}^*) \\ w_i &= \cos t, \end{aligned} \tag{17}$$

where $k > 0$. Considering N vehicles in the plane (see Fig. 1) over a graph G, the positions converge to the desired positions and the direction angles converge to zero if G is connected and realizable.

VII. Collision Avoidance

A. Objective

The objective is to propose a method that can prevent collisions between other agents and obstacles, while maintaining the formation. As a technical definition, collision avoidance should prevent point agents to simultaneously occupy the same point in space. Ultimately collision avoidance should be an expansion of the formation control. When obstacles are encountered, the goal is for the agents to remain in the formation, this means the formation seen as a unit or geometric structure may increase or decrease in size to avoid collisions while maintaining the geometric properties of the formation.

B. Potential Field

To implement a method to avoid collisions a repulsive potential field function V_{ij} is designed according to [6]. The function induces repulsive behaviour between agents i and $j \in M_i$ within a distance d_1 (where $0 < d_1 < d$ and d is the max sensing range for an agent), preventing collision.

Consider

$$V_{ij}(\beta_{ij}) = V_{ij}(||q_i - q_j||^2), \tag{18}$$

where β_{ij} is the distance between agents i and j as q indicates the position of an agent. V_{ij} now requires following properties:

- $V_{ij}(0)$ must be maximum value. If V_{ij} unbounded then $V_{ij}(0) \to \infty$
- Countinuously differentiable for all β_{ij}.
- $\frac{\partial V_{ij}}{\partial q_i} = 0$ and $V_{ij} = 0$ when $\beta_{ij} > d_1^2$
- The partial derivative $\rho_{ij} \triangleq \frac{\partial V_{ij}}{\partial \beta_{ij}}$ satisfies $\rho_{ij} < 0$ for $0 < \beta_{ij} < d_1^2$ and $\rho_{ij} = 0$ for $\beta_{ij} \geq d_1^2$

A design of the repulsive potential field function that satisfies the bound $|\rho_{ij}| \leq \frac{\rho}{\beta_{ij}}$ is given by

$$V_{ij}(\beta_{ij}) = \begin{cases} \rho \ln(\frac{1}{\beta_{ij}}) & \text{for } \beta_{ij} < c \\ h(\beta_{ij} - d^2)^2 & \text{for } c \leq \beta_{ij} < d^2 \\ 0 & \text{for } \beta_{ij} \geq d^2, \end{cases} \tag{19}$$

To ensure V_{ij} is continuously differentiable, c and h are chosen accordingly. This gives a partial derivative of V_{ij} as following

$$\rho_{ij}(\beta_{ij}) = \frac{\partial V_{ij}}{\partial \beta_{ij}} = \begin{cases} -\frac{\rho}{\beta_{ij}} & \text{for } \beta_{ij} < c \\ 2h(\beta_{ij} - d^2) & \text{for } c \leq \beta_{ij} < d^2 \\ 0 & \text{for } \beta_{ij} \geq d^2. \end{cases} \tag{20}$$

TABLE I
Simulation parameters for trajectory tracking

Parameter	Value
r_1	1
r_2	1
q_1	10
$\boldsymbol{Q}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
$\boldsymbol{P}_2(t=0)$	$\begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$
$\boldsymbol{u}_d(t)$	$\begin{bmatrix} 2, \frac{0.1\pi}{6}\cos 0.1t \end{bmatrix}^T$
$\boldsymbol{p}(t=0)$	$\begin{bmatrix} 0, 0, 0 \end{bmatrix}^T$
$\boldsymbol{p}_d(t=0)$	$\begin{bmatrix} 0, 5, \frac{\pi}{4} \end{bmatrix}^T$

Furthermore, the potential field controller for agent i is represented as

$$u_i = -2 \sum_{j \in M_i} \rho_{ij}(q_i - q_j). \tag{21}$$

As the potential field controller is defined, collision avoidance can now be integrated with formation control by merging the two controllers (14) and (21) as:

$$\boldsymbol{u} = -k[\boldsymbol{L}\boldsymbol{p}(t) + \boldsymbol{L}\dot{\boldsymbol{p}}(t)] - 2\boldsymbol{R}\boldsymbol{p}(t), \tag{22}$$

where the summations in (14) are rewritten using the Laplacian matrix and the repulsive matrix $\boldsymbol{R}$ is finally given as

$$\boldsymbol{R} = \begin{bmatrix} R_{11} & \dots & R_{1b} \\ \vdots & \ddots & \vdots \\ R_{a1} & \dots & R_{ab} \end{bmatrix}, \tag{23}$$

where the elements on the main diagonal in $\boldsymbol{R}$ are given by $R_{ii} = \sum_{j \neq i} \rho_{ij}$ and the other elements are given by $R_{ij} = -\rho_{ij}$.

VIII. Simulation

To perform the simulations the program MATLAB is used. The methods used to solve the first-order differential equations are the forward Euler method and MATLAB's own differential equations solver; ode-45.

A. Trajectory Tracking

Simulations were first done in the chained form given by equation (3). The following equation was used to transform back the results to x, y, θ-coordinates in order to plot them:

$$\theta = x_1, x = x_2 \sin x_1 + x_3 \cos x_1, y = -x_2 \cos x_1 + x_3 \sin x_1 \tag{24}$$

The error vector $\boldsymbol{x}_e$ is obtained by simulating the cascade system given by (8) and (9). Furthermore the desired trajectory vector $\boldsymbol{x}_d$ is obtained by simulating the kinematic model in chained form given by (5). Note $\boldsymbol{x}$ in (5) is replaced by the desired trajectory $\boldsymbol{x}_d$. Also, (4) is used to replace v_1 and v_2 in (5). Parameters from table I were used in both simulations.

In order to calculate the vehicle's actual trajectory, the following equation obtained from (6) is used:

$$\boldsymbol{x} = \boldsymbol{x}_e + \boldsymbol{x}_d \tag{25}$$

The actual trajectory and the desired trajectory are both shown in Fig. 2. The tracking error for the vehicle is shown in Fig. 3. The system is stabilized, i.e. the error is converging to zero, after approximately three seconds.

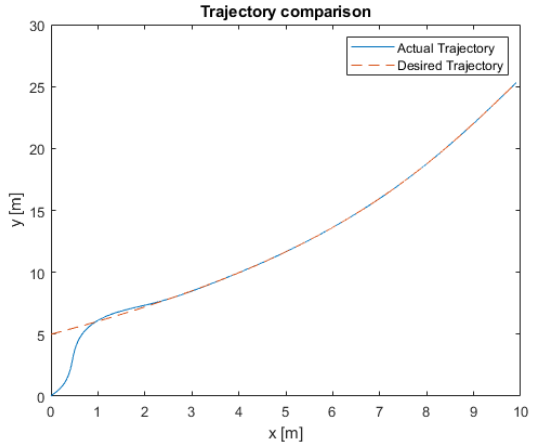


Fig. 2. An agent's actual trajectory compared to the desired trajectory.

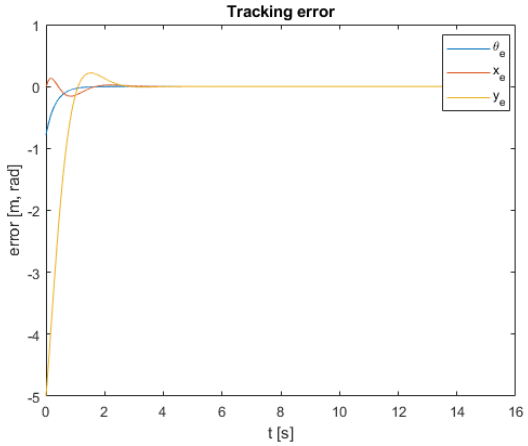


Fig. 3. The tracking error for the trajectory tracking controller.

B. Formation Control

1) Double-integrator model: In order to achieve the desired formation, target relative positions in the xy-plane and relative velocities are chosen. The positions of the multi-agent system are received by simulating the node dynamics described in (12) - (14) which according to [5] can be written as following

$$\begin{bmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{v}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I}_n \\ -\boldsymbol{L} & -\boldsymbol{L} \end{bmatrix} \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{v} \end{bmatrix}, \tag{26}$$

where $\boldsymbol{L}$ is the Laplacian matrix. It can be calculated by $\boldsymbol{L} = \boldsymbol{D}\boldsymbol{D}^T$ and $\boldsymbol{D} = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ 1 & 0 & 0 & -1 \end{bmatrix}$ is the incidence matrix of the graph G in Fig. 4.

The initial and final formation of the multi-agent system and their trajectories are shown in Fig. 5. Relative position between the agents can be seen in Fig. 6.

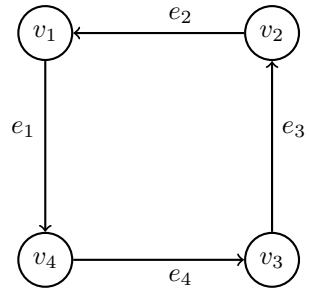


Fig. 4. Graph G of the multi-agent system with nodes v_i and edges e_i.

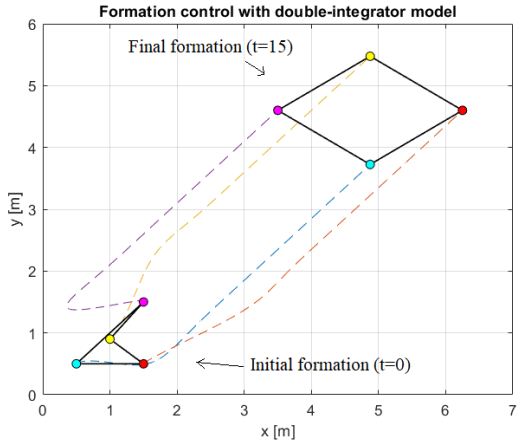


Fig. 5. Trajectories of each agent in the xy-plane from initial formation to final formation in the time window t=0s to t=15s, using the double-integrator model.

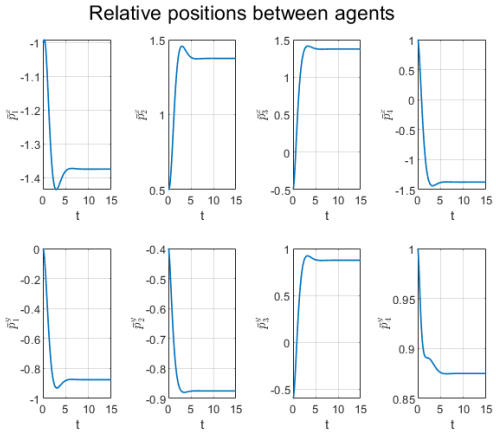


Fig. 6. Relative positions between agents in the time window t=0 to t=15 in Fig. 5. See end of discussion.

2) Nonholonomic model: Simulating (17) gives the control variables v_i and w_i which are then inserted into (16). The result of the final formation with nonholonomic agent model is seen in Fig. 7.

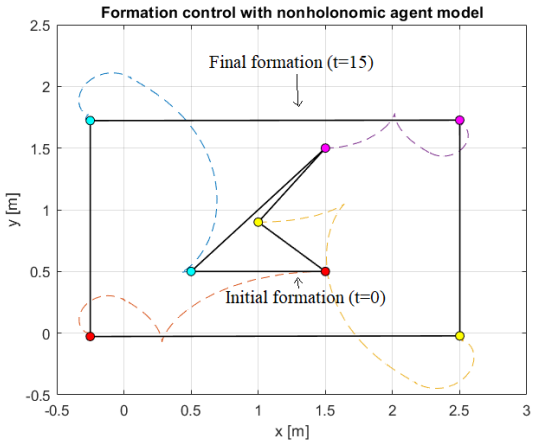


Fig. 7. Trajectories of each agent in the xy-plane from initial formation to final formation in the time window t=0s to t=15s, using the nonholonomic model.

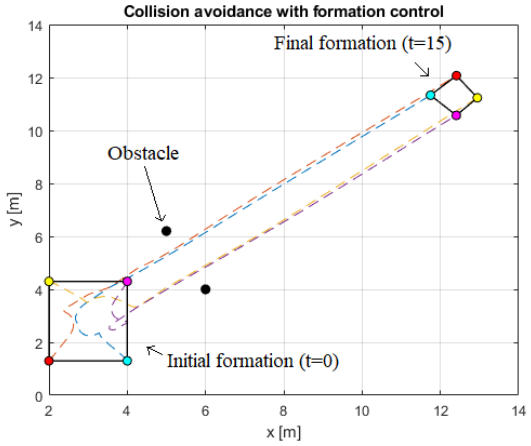


Fig. 8. Trajectories of each agent in the xy-plane from initial formation to final formation in the time window t=0s to t=15s. With two static obstacles.

TABLE II
SIMULATION PARAMETERS FOR COLLISION AVOIDANCE

Parameter	Value
ρ	$8/e^2$
c	1/e
d^2	3/e
h	2
$[x_1, y_1]^T_{target}$	$[4, 3]^T$
$[x_2, y_2]^T_{target}$	$[3, 3]^T$
$[x_3, y_3]^T_{target}$	$[3, 4]^T$
$[x_4, y_4]^T_{target}$	$[4, 4]^T$
$[x_5, y_5]^T_{object1}$	$[5, 6.2]^T$
$[x_6, y_6]^T_{object2}$	$[6, 4]^T$

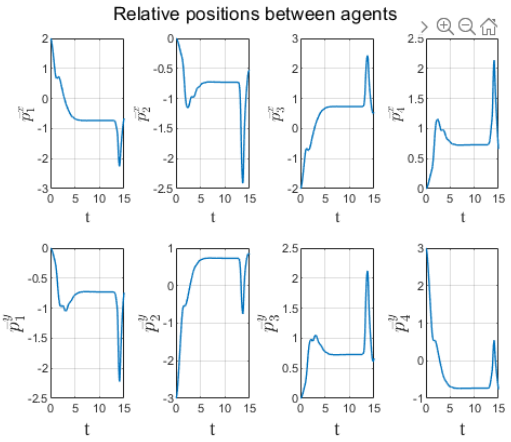


Fig. 9. Relative positions between agents in the time window t=0s to t=15s in Fig. 8. See end of discussion.

C. Collision Avoidance

The simulations for collision avoidance are an expansion on the simulations for formation control with the double-integrator model. Two obstacles are added, functioning as static agents. To add two obstacles, two new columns are added in the $\boldsymbol{R}$ matrix and the expressions on the main diagonal are updated. The incidence matrix $\boldsymbol{D}$ stays unchanged.

The parameters in table II; ρ, c, d^2 and h are all used to calculate the repulsiveness expressions, these expressions represent the elements in the $\boldsymbol{R}$ matrix. Furthermore, the target vectors in table II represents the desired relative positions.

The trajectories of the multi-agent system with collision avoidance and formation control are seen in Fig. 8. Relative positions between the four agents are shown in Fig. 9. Fig. 10 and 11 shows each agent's x-position and y-position over time, respectively.

IX. DISCUSSION

A. Trajectory Tracking

Designing a stable trajectory tracking controller for the differential-drive vehicle in model (2) was achieved, see Fig. 2. Asymptotic stability was conformed in Fig. 3 since the tracking errors converged to zero after three seconds. There was a small overshoot in two of the graphs in Fig. 3. It can be eliminated by equating the initial position of the desired and actual trajectory. This can be done assuming the vehicle is in an ideal environment. However in real-life application this assumption is not valid.

B. Formation Control

The objective of proposing a formation controller was fulfilled. The formations followed the geometric shape of a parallelogram as Fig. 5 and 7 show. Considering Fig. 5 it can be seen that the initial formation rearranged itself and formed a formation of a parallelogram after some time. The reason the final formation is located at increased x and y positions is because each agent needed to be initialized with some velocity in the double-integrator model. This is in contrast to Fig. 7 where the final formation is still centralized in roughly the same position as the initial formation. This is because with the nonholonomic model does not need to initialize velocity, thus agents can be seen to converge to a velocity of zero. Note that the agents are still moving to reach the final formation but the center point remains unchanged.

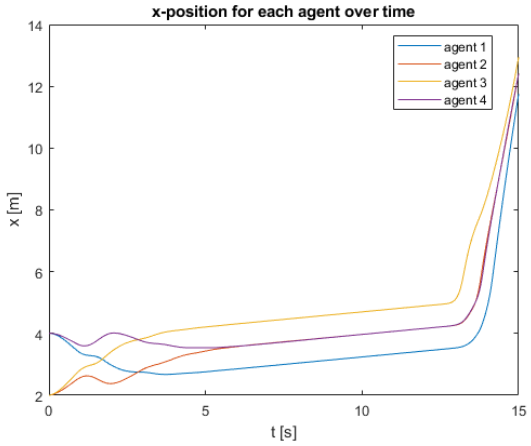


Fig. 10. x-position over time for all agents from t=0s to t=15s in Fig. 8

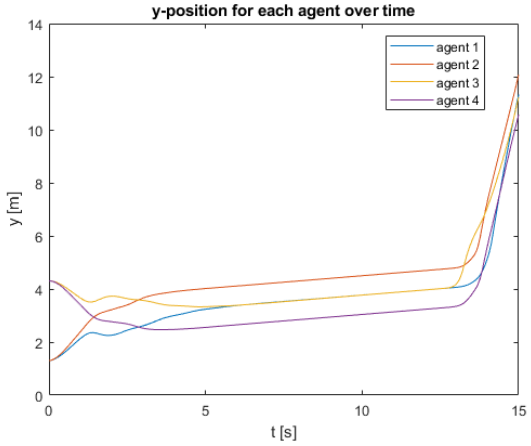


Fig. 11. y-position over time for all agents from t=0s to t=15s in Fig. 8.

C. Collision Avoidance

The goal of designing a collision avoidance controller was reached and the main result can be seen in Fig. 8. The collision avoidance controller (22) was as mentioned in section VII a combination between the double-integrator formation controller presented in (14) and the potential field controller proposed in (21). Due to time constraints collision avoidance was not extended with the nonholonomic model. In Fig. 8 the initial formation successfully minimized its formation structure to get through the obstacles without collision between agents nor obstacles. The same figure also shows the trajectories of each agent. Despite trajectories crossing between multiple agents it does not mean a collision occurred between them, because they can occupy different positions at different times. The time variable is not apparent in Fig. 8. Instead Fig. 10 and 11 prove that there were no collisions by plotting the position of each agent over time. For a collision to happen the x- and y-value for two or more agents have to be equal at same point in time. The figures show that even if some agents share the same x-coordinate at a point in time, they never at the same time share the equivalent y-coordinate and vice verse. Fig. 10 and 11 also show that the agents move with a higher velocity after roughly 13.5 seconds, this is revealed by the slope of the graph. The reason for this is that at that point in time they bypassed the obstacles, meaning the repulsive forces from the obstacles started pushing them away in the same direction as their general velocity were pushing them, hence the increase in speed. This also explains why the agents are moving slower when approaching the obstacles, because in that case the general velocity of the agents were fighting against the repulsive forces from the obstacles. In a practical sense the algorithm simulates a human's driving well, cautious when approaching obstacles to avoid accidents and less passive when leaving an obstacle. Fig. 6 and 9 explains the relative positions between agents over time unit seconds. p_1^x shows relative position in x direction for e_2, p_2^x for e_3, p_3^x for e_4 and p_4^x for e_1. Same for all p_i^y.

X. Conclusion

The project looked at cooperative control of autonomous ground vehicles. It divided the problem into three parts, trajectory tracking, formation control and collision avoidance. Firstly a trajectory controller was designed, the tracking error was shown to converge to zero after roughly three seconds, this meant the actual trajectory would converge with the desired trajectory in the same amount of time the error converged to zero. A formation controller was then proposed, using displacement-based formation control and looking at two different models, the double-integrator and the nonholonomic model. The results showed that for both methods an initial formation, that of a non parallelogram, would converge to a final formation of a parallelogram within a reasonable time frame for both models. Finally a repulsive potential field controller was integrated to the double-integrator formation controller to create the final collision avoidance controller that included formation control. The results were successfully showing that the shape of the formation was reserved while the size of it changed to avoid obstacles in the plane. At the same time it was also shown that no agents collided during the process. In essence the objective of the project was fulfilled with successful results.

Acknowledgment

Many thanks to project supervisor Fei Chen for his assistance, direction and patience throughout the project.

References

[1] L. Mora, X. Wu, and A. Panori, "Mind the gap: Developments in autonomous driving research and the sustainability challenge," *Journal of Cleaner Production*, vol. 275, p. 124087, Dec. 2020.

[2] J. M. Anderson, N. Kalra, K. D. Stanley, P. Sorensen, C. Samaras, and T. A. Oluwatola, *Autonomous Vehicle Technology: A Guide for Policymakers*. Santa Monica, CA: RAND Corporation, 2016.

[3] Z. Qu, *Cooperative Control of Dynamical Systems Applications to Autonomous Vehicles*. London, United Kingdom: Springer-Verlag, 2009.

[4] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, Oct. 2015.

[5] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*, stu - student edition ed. Princeton University Press, 2010.

[6] D. V. Dimarogonas and K. J. Kyriakopoulos, "Connectedness preserving distributed swarm aggregation for multiple kinematic robots," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1213–1223, Oct. 2008.

Multi-Robot Motion Planning With Control Barrier Functions for Signal Temporal Logic Tasks

Cecilia Brage and Johanna Johansson

***Abstract*—Autonomous robots have the potential to accomplish a wide variety of assignments. For this to work in reality, the robots need to be able to perform specific tasks while safety for both them and their environment is ensured. Signal temporal logic (STL) was used to define timed tasks for the agents to perform and control barrier functions (CBFs) were used to design a controller for their movements. In this paper, a set of STL tasks were considered, which two robots were instructed to satisfy in a simulation of a warehouse environment. The two agents started next to each other, then the set of tasks instructed them to move to two separate areas, then meet up again and move in a formation back towards their starting area. Control barrier functions were employed to ensure the satisfaction of the set of STL tasks. The agents designed their actions towards satisfying the given tasks without considering a safety distance to the other robot at first. To later ensure safety, a collision avoidance mechanism was introduced. The scenario without collision avoidance proved more effective paths for the agents. They moved to satisfy the tasks with less disturbance than the scenario where collision avoidance was considered. However, the scenario with the collision avoidance mechanism proved successful and the agents satisfied their tasks without colliding with each other.**

***Sammanfattning*—Autonoma robotar har potential att utföra en stor mängd olika uppgifter. För att detta ska fungera i verkligheten, behöver robotarna kunna genomföra specifika uppgifter medans både deras egen och omgivningens säkerhet är säkerställd. Signal temporal logic (STL) användes för att definiera tidsinställda uppgifter åt robotarna att utföra och control barrier functions (CBFs) användes för att designa en controller för deras rörelser. I den här rapporten betraktades en uppsättning av STL-uppgifter, vilka två robotar instruerades att uppfylla i en simulering av en lagermiljö. De två robotarna startade bredvid varandra, sen instruerade STL-uppgifterna dem att röra sig till två separata områden, sen mötas upp igen och röra sig i formation tillbaka mot sitt startområde. Control barrier functions användes för att garantera uppfyllandet av STL-uppgifterna. Robotarna anpassade sina rörelser till att uppfylla de givna uppgifterna, först utan hänsyn till någon säkerhetsmarginal till den andra roboten. För att senare garantera säkerhet introducerades en extra mekanism för att undvika kollision. Scenariot utan att undvika kollision visade på effektivare rörelsebanor hos robotarna. De rörde sig mot att uppfylla uppgifterna med färre störningar än scenariot då kollision aktivt undveks. Scenariot med mekanismen för att undvika kollision visade sig dock framgångsrikt och robotarna uppfyllde sina uppgifter utan att kollidera med varandra.**

***Index Terms*—autonomous robots, autonomous systems, signal temporal logic, control barrier function, formation control, collision avoidance, formal methods.**

Supervisor: Maria Charitidou

TRITA number: TRITA-EECS-EX-2021:143

I. Introduction

The significance of autonomous robots performing tasks in our society has increased notably with their various areas of application being continuously extended. Over the years, autonomous robots have been considered in a variety of tasks, for example, according to [1], these robots can now perform different tasks such as cleaning floors, apple-picking and different kinds of medical assignments. The reason for employing a robot for these types of tasks is mainly to make production and maintenance more efficient [1]. However, an example of a recent medical task, is autonomous robots disinfecting hospitals from coronavirus and other harmful particles [2], which is a repetitive and potentially dangerous assignment for a human to perform. Another example of use is inside warehouses, where robots stock, move and deliver goods. This requires the robotic agent to perform a given task, while simultaneously keeping track of its surroundings.

To achieve this, a reliable and broad framework needs to be provided in order for the robots to perform all kinds of tasks. Such a framework would guarantee safety for the robots and their environment, meaning all collisions would be avoided, while providing the possibility of application to a wide variety of tasks. In this project, two robots were instructed to move to their own designated goal area, after that meet up again and move in a formation back to the starting area. In conjunction to the aforementioned tasks, the agents needed to stay within their workspace.

In this paper we considered two different scenarios. In the first one, the agents designed their actions towards satisfaction of the given STL task. In the second scenario, a collision avoidance mechanism was also introduced, ensuring the agents safety during their mission. The entire workspace will be defined in III. *B. Workspace decomposition*. For this project, no static obstacles were be used, since the main focus was the task of reaching a specified area, formation control and collision avoidance between agents. When a cluttered environment is considered, obstacle avoidance can be ensured using time-invariant control barrier functions [3]. Nevertheless, this is beyond the scope of this paper and could be considered as a subject of future study.

The motion plan was created and designed with signal temporal logic (STL) and control barrier functions (CBFs). STL was used to express the tasks the autonomous robots were to perform, using logical operators. Then, a controller was designed using control barrier functions. The controller determined the agents' next actions towards satisfying the STL tasks. The analysis was made in continuous time, however, a

sampling period was used in the simulation while calculating the movements through time based on the given tasks, predicates and their robustness requirements. All methods and concepts will be explained further in the next section.

II. Background

Here follows a theoretical background on the concepts used in the project as well as a summary of previous work done related to the area. Please note that the descriptions in this section are conceptual, the mathematical definitions will be provided in III. Mathematical Framework or IV. Method.

A. Theoretical Background

The theoretical background contains explanations for the most important concepts and terms used in this paper. Some concepts also contain mathematical definitions, which will be thoroughly explained in subsequent sections.

1) Signal temporal logic tasks: In this paper, the tasks the agents were set to perform will be described using signal temporal logic (STL). The word temporal is here referring to the logic's relation to time (see [4] for definition) indicating that the agents' movements will progress chronologically over time. This mathematical language can be compared to linear temporal logic (LTL), which requires discretization of the signal. With STL, a signal continuous over time can be used combined with robustness semantics (will be explained in II.*A. 3)*). Another advantage of STL is that strict deadlines can be set for the tasks. For example, if a task should be active from time *a* to *b*, the task will be defined in a conceptual as

$$\text{task} := \|\text{description}\|_{[a,b],\text{norm}} \leq \text{requirement}. \tag{1}$$

The description will here tell which agent the task concerns and what kind of task said agent should perform, for example keep a certain distance from or reach a specified area. The requirement will in this case describe the maximum or minimum distance the agent should keep to the specified area's center point. The norm describes the shape of the specified area, for example, the 1-norm will create a diamond shape, the 2-norm a perfect circle and the ∞-norm a square. Examples of how the tasks are written mathematically are shown in III. *A. STL Syntax and Semantics Over Motion Trajectory*.

2) Control Barrier Functions: For the agents to iteratively try to satisfy their tasks in a reasonable way, a controller needs to be created. This controller is designed to satisfy the tasks within their deadline, while making sure the tasks which are always active stay satisfied through the entire simulation. A control barrier function, defined in [5], ensures the constantly active tasks to be satisfied with complete forward invariance. Full definition of forward invariance can be found in [6].

This paper will explore the possibility for control barrier functions to be used for reaching an area and formation control combined. The controller is based on a barrier function $\mathfrak{b}(\mathbf{x}, t)$ which much be positive for all the tasks to be satisfied for all $t \geq 0$. Mathematical explanations of control barrier functions will be provided in IV. *Control Barrier Functions*.

3) Robustness: To be able to evaluate how well an STL task is satisfied, or violated, robustness semantics, defined in III. *C. Robustness Semantics*, can be introduced. In [7] it is explained that STL taks (with robustness semantics) offer a broader indication of the satisfaction of the tasks compared to linear temporal logic (LTL) tasks, where the satisfaction only can be determined to be true or false. The robustness semantics are, as [7] further explains, an under-approximation of a more detailed version of robustness, the robustness degree. The robustness semantics are less computationally demanding than the robustness degree and will therefore be used in this paper. Further explanation of robustness can be found in [7].

To put robustness in the perspective of this project, the controller should aim for every task to be satisfied with the highest robustness possible. Without robustness, the task will either be entirely satisfied or it will not be satisfied at all. For example, if an agent is assigned a task to reach a specific area, it will be satisfied as soon as the agent reaches the edge of the area (the fact that the agent has reached the area is true). However, with robustness, the controller will aim to iteratively calculate a path for the agent to approach the center of the area. This is because the robustness will be even higher the closer to the center of the area the agent moves. The controller will aim for the highest robustness possible for all tasks, which means the most optimal path will be calculated.

4) Temporal Behaviour: The temporal behaviour function $\gamma(t)$ is used to guarantee satisfaction of the local task ϕ with a certain robustness. By adjusting the temporal behaviour for a task the time controlled part of the system changes, leading to a difference in when in time and with which robustness the task is satisfied. Furthermore, γ_0 is defined as the initial value for the γ function, $\gamma_0 = \gamma(t = 0)$ and γ_∞ is the value for the function when the time t is greater than the time limit, t^*, for when the task must have been satisfied, $\gamma_\infty = \gamma(t \geq t^*)$. A condition for the temporal behaviour is that γ_∞ must be greater than or equal to the desired robustness value r.

5) Formation Control: For autonomous robots to be of use in every kind of warehouse environment, they might be required to work together to perform certain assignments. For example, if a large and/or heavy object has to be moved by more than one robot, tasks would have to be formulated to keep the agents close to each other. These are called formation tasks. Formation control has the agents maintain a certain distance to each other [8] and may sometimes demand them to move in a specified geometric shape or reach a specific goal point. Only a part of this project contained formation control did not have the requirement of a specific shape for the formation.

6) Collision Avoidance: To guarantee the safety for the agents, a collision avoidance mechanism was utilized. This controller is usually implemented as a term in the dynamics for the robots as seen in [6], where the mechanism is a function of the distance between the agents and can be tuned in with a constant. The purpose of the controller is to prevent the agents from colliding when they get too close to each other.

B. Earlier Work

Earlier work provides a summary of the previous work done within the area or closely related to it.

In [6], an efficient method to design a controller ensuring the satisfaction of the STL tasks using control barrier functions is introduced. The barrier function of the conjunction of the formulas is designed as an under-approximation of the minimum of the barrier functions corresponding to the sub-formulas. This approximation was also used in this project for efficiency, since both STL and control barrier functions were used here as well.

In [9] the agents are exposed to conflicting tasks, where the tasks are instead given priorities. STL and receding horizon optimization are used to describe the temporal and collision avoidance tasks. This paper also considers collision avoidance by introducing a collision avoidance mechanism in the dynamics of the system, however the receding horizon optimization will be exchanged for control barrier functions. In general, the receding horizon optimization is more computationally intensive than control barrier functions. In the approach used in this paper, a computationally efficient convex quadratic program (QP) is solved. The STL tasks considered in this project were assumed to be non-conflicting. This implies that there always exists a trajectory $\mathbf{x}$ such that $\mathbf{x} \models \phi$, where $\mathbf{x}$ describes the states of all agents, i.e. their positions in the workspace.

STL is used in [10] to describe tasks and mixed integer-linear constraints for optimization for controlling climate and energy inside a building. While this scenario differs from this project, since this project investigated robot control in a warehouse, [10] uses robustness for satisfaction of the STL formulas, which has also been used in this paper.

The problem with conflicting STL defined tasks for systems with a number of agents and how to define a controller based on control barrier functions is explored in [3]. Through online collaboration amongst the agents when encountering conflicting tasks, the system can find the least violating solution. In this scenario, the agents were not exposed to conflicting tasks, however, they needed to move close to each other while respecting a minimum safe distance.

III. Mathematical Framework

A. STL Syntax and Semantics Over Motion Trajectory

Signal temporal logic contains predicates μ_i which are vital conditions that the system should satisfy at all times. For example a predicate could be that an agent should stay within a defined workspace during a simulation session. The predicates are of a Boolean nature, either true or false,

$$\mu_i = \begin{cases} \top \text{ (True)}, & h_i(\mathbf{x}) \geq 0 \\ \bot \text{ (False)}, & h_i(\mathbf{x}) < 0 \end{cases} \tag{2}$$

based on a predicate function h_i, where $i = 1, ..., N$ and N is the total number of predicates for the system. A positive value for the predicate function implies that the predicate has been satisfied.

The syntax for STL is defined by [6] as

$$\phi := \top \mid \mu \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \mathcal{U}_{[a,b]} \phi_2 \tag{3}$$

where ϕ_1 and ϕ_2 describes the STL formulas and $\mathcal{U}$ is the Until operator.

The semantics for the STL are defined in [6] as following for a signal $\mathbf{x}$ at the time t. The predicate μ is satisfied when the predicate function $h(\mathbf{x}(t))$ is greater than or equal to zero as in

$$(\mathbf{x}, t) \models \mu \Leftrightarrow h(\mathbf{x}(t)) \geq 0. \tag{4}$$

The task ϕ is not satisified if the system does not accomplish it for the signal $\mathbf{x}$ at the time t as in

$$(\mathbf{x}, t) \models \neg\phi \Leftrightarrow \neg((\mathbf{x}, t) \models \phi). \tag{5}$$

The tasks ϕ_1 and ϕ_2 are satisfied if both of them are accomplished at the time t as in

$$(\mathbf{x}, t) \models \phi_1 \wedge \phi_2 \Leftrightarrow (\mathbf{x}, t) \models \phi_1 \wedge (\mathbf{x}, t) \models \phi_2. \tag{6}$$

Definition of the Future (Eventually) operator meaning that the task ϕ should be satisfied at some point in the time interval $[a, b]$ as in

$$(\mathbf{x}, t) \models \mathcal{F}_{[a,b]}\phi \Leftrightarrow \exists t_1 \in [t + a, t + b] \text{ s.t. } (\mathbf{x}, t_1) \models \phi. \tag{7}$$

Definition of the Global (Always) operator meaning that the task ϕ should be satisfied at all times in the time interval $[a, b]$ as in

$$(\mathbf{x}, t) \models \mathcal{G}_{[a,b]}\phi \Leftrightarrow \forall t_1 \in [t + a, t + b], (\mathbf{x}, t_1) \models \phi. \tag{8}$$

Definition of the Until operator meaning that during the time interval $[a, b]$ the task ϕ_1 should be satisfied at all times until $t = t_1$ when task ϕ_2 should be satisfied at some point as in

$$(\mathbf{x}, t) \models \phi_1 \mathcal{U}_{[a,b]} \phi_2 \Leftrightarrow \exists t_1 \in [t + a, t + b] \text{ s.t.}$$
$$(\mathbf{x}, t_1) \models \phi_2 \wedge \forall t_2 \in [t, t_1], (\mathbf{x}, t_2) \models \phi_1. \tag{9}$$

The Until operator can as stated in [11] be written as a conjunction of a Global operator and a Future Operator as in

$$\phi_1 \mathcal{U}_{[a,b]} \phi_2 = \mathcal{G}_{[a,t_1]}\phi_1 \wedge \mathcal{F}_{[t_1,t_1]}\phi_2. \tag{10}$$

In this project the STL fragment, as defined in [6], is considered as

$$\psi := \top \mid \mu \mid \neg\mu \mid \psi_1 \wedge \psi_2 \tag{11}$$

$$\phi := \mathcal{G}_{[a,b]}\psi \mid \mathcal{F}_{[a,b]}\psi \mid \psi_1 \mathcal{U}_{[a,b]}\psi_2 \mid \phi_1 \wedge \phi_2 \tag{12}$$

where ψ_1, ψ_2, ϕ_1 and ϕ_2 are formulas of classes ψ (11) and ϕ (12) respectively.

1) Robot Model: The state for the robots i.e. their positions are defined as

$$\mathbf{x} = \begin{bmatrix} p_{1,x} & p_{1,y} & p_{2,x} & p_{2,y} \end{bmatrix}^T \tag{13}$$

where $p_1 = [p_{1,x} \ p_{1,y}]$ represents the x-coordinate and y-coordinate for agent 1. The same applies for agent 2 whose position is $p_2 = [p_{x,2} \ p_{2,y}]$.

The dynamics i.e. the velocities for the agents are defined as

$$\dot{\mathbf{x}} = f(\mathbf{x}) + \mathbf{u} \tag{14}$$

where $\mathbf{u} = [u_{1,x} \ u_{1,y} \ u_{2,x} \ u_{2,y}]^T$ contains the control inputs for the system. $f(\mathbf{x})$ is the term for the collision avoidance and is introduced in the system according to [6] as

$$f_{i,x} = \sum_{j=1, j\neq i}^{M} k_i \frac{p_{i,x} - p_{j,x}}{\|p_i - p_j\| + 0.000001} \tag{15}$$

$$f_{i,y} = \sum_{j=1, j\neq i}^{M} k_i \frac{p_{i,y} - p_{j,y}}{\|p_i - p_j\| + 0.000001} \tag{16}$$

where k_i is a constant greater than 0 and $M = 2$ for the total number of agents.

The two agents are of cylindrical shape with the radius 0.15 meters.

B. Temporal Constraint Expressed in STL

The temporal behaviour function γ can either be exponential or linear. [11] states that the linear function is preferred over the exponential function since the latter may make $\mathbf{u}$ larger than desired. Their definition of the γ function for temporal behaviour is

$$\gamma_j(t) = \begin{cases} \frac{\gamma_{j,\infty} - \gamma_{j,0}}{t_j^*} t + \gamma_{j,0} & \text{if } t < t_j^* \\ \gamma_{j,\infty} & \text{if } t \geq t_j^* \end{cases}. \tag{17}$$

In [12] the exponential function is used and defined by

$$\gamma_j(t) = (\gamma_{j,0} - \gamma_{j,\infty}) e^{-\mathfrak{l}_j t} + \gamma_{j,\infty} \tag{18}$$

where

$$\mathfrak{l}_j = \frac{-\ln(\frac{r - \gamma_{j,\infty}}{\gamma_{j,0} - \gamma_{j,\infty}})}{t_j^*} \tag{19}$$

and r is the value of the desired robustness. t_j^* is defined differently depending on whether the task is a Future or Global task. For a task ϕ defined for a time interval $[a, b]$ the following applies

$$t_j^* = \begin{cases} b_j, & \text{if } \mathcal{F}_{[a_j,b_j]}\phi \\ a_j, & \text{if } \mathcal{G}_{[a_j,b_j]}\phi \end{cases}. \tag{20}$$

For an Until operator $\phi_1 \mathcal{U}_{[a_j,b_j]} \phi_2$, which can be written as a conjunction of a Global task and a Future task as in (10), (20) can be used to obtain t_j^* for each subtask.

C. Robustness Semantics

The semantics for robustness $\rho(\mathbf{x}, t)$ are defined by [13] as

$$\rho^{\mu}(\mathbf{x}, t) = h(\mathbf{x}(t)) \tag{21}$$

$$\rho^{\neg\phi}(\mathbf{x}, t) = -\rho^{\phi}(\mathbf{x}, t) \tag{22}$$

$$\rho^{\phi_1 \wedge \phi_2}(\mathbf{x}, t) = \min(\rho^{\phi_1}(\mathbf{x}, t), \rho^{\phi_2}(\mathbf{x}, t)) \tag{23}$$

$$\rho^{\phi_1 \mathcal{U}_{[a,b]} \phi_2} = \max_{t_1 \in [t+a, t+b]} \min\left(\rho^{\phi_2}(\mathbf{x}, t_1), \min_{t_2 \in [t, t_1]} \rho^{\phi_1}(\mathbf{x}, t_2)\right) \tag{24}$$

$$\rho^{\mathcal{F}_{[a,b]}\phi}(\mathbf{x}, t) = \max_{t_1 \in [t+a, t+b]} \rho^{\phi}(\mathbf{x}, t_1) \tag{25}$$

$$\rho^{\mathcal{G}_{[a,b]}\phi}(\mathbf{x}, t) = \min_{t_1 \in [t+a, t+b]} \rho^{\phi}(\mathbf{x}, t_1). \tag{26}$$

D. Problem Definition

The aim of this project is to derive a control input $\mathbf{u}(\mathbf{x}, t)$ so that tasks based on the syntax described in (3) and the fragment described in (11) and (12) are satisfied within the specified time constraints. This while simultaneously keeping the barrier constantly positive during the entire simulation. The simulations created in this paper are set to last 25 seconds, which is the total duration of the task for the system.

IV. Method

A. Control Barrier Functions

The barrier function for each predicate function h_i is calculated by [11] as

$$\mathfrak{b}_i(\mathbf{x}, t) = -\gamma_j(t) + h_i(\mathbf{x}). \tag{27}$$

When calculating the total barrier $\mathfrak{b}(\mathbf{x}, t)$ for the entire system, an under-approximation of the minimum value of $\mathfrak{b}_i$ was used as stated in [6], to enable usage of the STL syntax. The barrier function $\mathfrak{b}(\mathbf{x}, t)$ is a smooth under-approximation of the minimum operator. The minimum operator is a non-differentiable function in general, which results in the gradient not being definable. An alternative is non-smooth analysis, however, the analysis becomes more complex. Therefore, an under-approximation that is differentiable in $\mathbf{x}$ is used if $h(\mathbf{x})$ is differentiable. The total barrier is defined as

$$\mathfrak{b}(\mathbf{x}, t) = -\ln\left(\sum_{i=1}^{p} \mathfrak{o}_i \exp(-\mathfrak{b}_i(\mathbf{x}, t))\right) \tag{28}$$

where $\mathfrak{o}_i$ is a variable, called the deactivation policy, defined in [12] discarding $\mathfrak{b}_i$ for the corresponding task once that task is satisfied, i.e. $t \geq t_j^*$ as in

$$\mathfrak{o}_i = \begin{cases} 1, & \text{if } t < t_j^* \\ 0, & \text{if } t \geq t_j^* \end{cases}. \tag{29}$$

The feedback controller $\mathbf{u}$ is described as

$$\min_{\mathbf{u}} \mathbf{u}^T \mathbf{u} \tag{30}$$

$$\frac{\partial \mathfrak{b}}{\partial \mathbf{x}}(f(\mathbf{x}) + \mathbf{u}) + \frac{\partial \mathfrak{b}}{\partial t} \geq -\alpha(\mathfrak{b}(\mathbf{x}, t)) \tag{31}$$

where the gradients are calculated as

$$\frac{\partial \mathfrak{b}}{\partial \mathbf{x}} = \frac{\sum_{i=1}^{p} \mathfrak{o}_i \exp(-\mathfrak{b}_i(\mathbf{x}, t)) \frac{\partial \mathfrak{b}_i}{\partial \mathbf{x}}}{\sum_{i=1}^{p} \mathfrak{o}_i \exp(-\mathfrak{b}_i(\mathbf{x}, t))} \tag{32}$$

$$\frac{\partial \mathfrak{b}}{\partial t} = \frac{\sum_{i=1}^{p} \mathfrak{o}_i \exp(-\mathfrak{b}_i(\mathbf{x}, t)) \frac{\partial \mathfrak{b}_i}{\partial t}}{\sum_{i=1}^{p} \mathfrak{o}_i \exp(-\mathfrak{b}_i(\mathbf{x}, t))}. \tag{33}$$

The feedback controller is based on control barrier functions which guarantee feasibility and solutions in continuous-time. It also provides convex problems which are easy to solve.

B. Workspace Decomposition

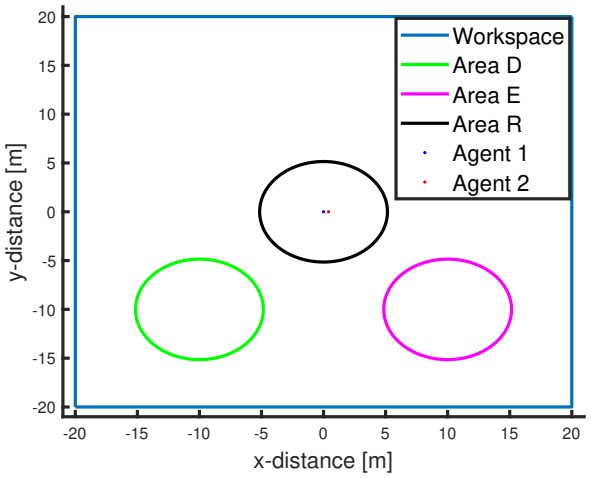


Figure 1. Map of the workspace with three circular areas for the agents to reach. The center circle indicating area R, the bottom left circle indicating area D and the bottom right circle indicating area E. The starting positions of agent 1 (left dot) and agent 2 (right dot) are indicated at the center of the picture.

The initial positions for the agents at $t = 0$ seconds were $p_1 = [0\ 0]$ and $p_2 = [0.4\ 0]$.

The workspace is shown in Fig. 1, showing the agents' initial positions, the three areas to reach and the walls of the warehouse. To instruct the agents to stay within their designated workspace, a square with a width and height of 40 meters, the predicate for the system was defined as

$$\mu_1 := \|p_n - p_A\|_\infty \leq \varepsilon. \tag{34}$$

Here, $n = 1, 2$ and A is the center of the workspace, p_A in $[0, 0]$. $\varepsilon = 19.85$ meters and is the maximum distance from the point A which the center points of the robots were allowed to go. This together with the agents' radiuses of 15 centimeters give a total workspace with the area 40x40 meters. The task to ensure the agents stay within their workspace is defined as

$$\phi_{env} = \wedge_{n=1}^{2} \phi_n \tag{35}$$

where

$$\phi_n = \mathcal{G}_{[0,25]} \mu_1. \tag{36}$$

To ensure continuous derivability the ∞-norm was rewritten into four linear predicate functions, two for the x-direction and two for the y-direction, defined as

$$h_i(\mathbf{x}) = \begin{cases} h_1 = \varepsilon - p_x \\ h_2 = \varepsilon + p_x \\ h_3 = \varepsilon - p_y \\ h_4 = \varepsilon + p_y \end{cases}. \tag{37}$$

where h_1 and h_2 verifies that the distance of the agents from the y-axis is less than ε and h_3 and h_4 verifies that the distance from the x-axis is less than ε.

C. Combining With Task Specifications

The system had a total of three tasks to satisfy during the simulation. Two of them were Future tasks, defined in (7), meaning that they would be satisfied eventually during a set interval in time. The last task was an Until task, defined in (9) and consists of two subtasks. During a set interval in time, one task should be satisfied until the other task is satisfied.

The first Future task instructed agent 1 to eventually during the time interval $[0, 10]$ seconds reach the checkpoint area D, a circle with a radius of 5.15 meters and its center point in $[-10; -10]$. Task 1 is described as

$$\phi_1 := \mathcal{F}_{[0,10]} \|p_1 - p_D\|_2 \leq d_1 \tag{38}$$

where $d_1 = 5$ meters is the largest, allowed distance for the agent to keep between its center point and point D which makes the predicate function

$$h_5(p_1) = d_1 - \|p_1 - p_D\|_2. \tag{39}$$

For the second Future task agent 2 was instructed to eventually reach another checkpoint area E during the time interval $[0, 10]$ seconds. It has its center point in $[10; -10]$ and is also a circle with a radius of 5.15 meters. Task 2 is described as

$$\phi_2 := \mathcal{F}_{[0,10]} \|p_2 - p_E\|_2 \leq d_1. \tag{40}$$

The maximum distance for the agent and the check point area is the same as for the previous task, $d_1 = 5$ meters and gives the predicate function

$$h_6(p_2) = d_1 - \|p_2 - p_E\|_2. \tag{41}$$

The final task, which is the Until task, is dependent on two subtasks, ϕ_3 and ϕ_4. ϕ_3 is a Global task that should be satisfied until task ϕ_4 is satisfied sometime in the future during the time interval $[15, 25]$ seconds. In this case the agents, 1 and 2, were instructed to remain in a formation until agent 2 reached area R. The Until task is described as

$$\phi_3 \mathcal{U}_{[15,25]} \phi_4. \tag{42}$$

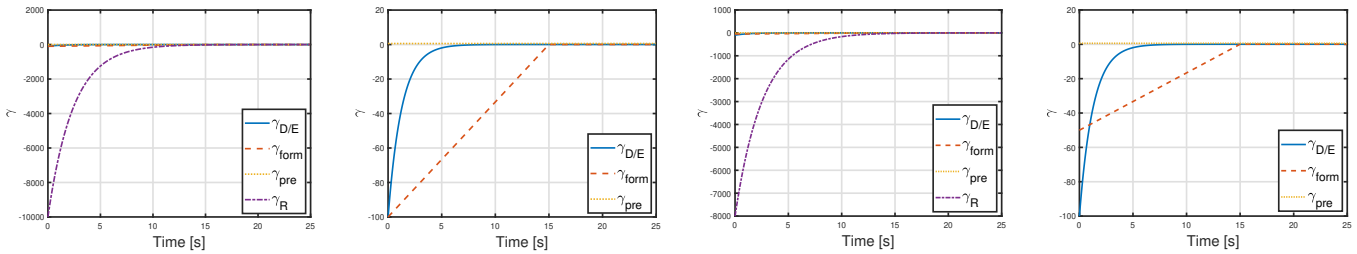
The first subtask, ϕ_3, is the task instructing the agents to stay in a formation. ϕ_{3a} is for the scenario with no safety guarantees and ϕ_{3b} is for the scenario considering the collision avoidance mechanism. The maximum distance between the agents for the formation in ϕ_{3a} was set to 1.3 meters and 2.3 meters for ϕ_{3b}. The reason for this was because the collision avoidance mechanism instructed the agents to stay away from each other to avoid collision. When the formation task was introduced simultaneously, the allowed distance between the agents in the formation task was increased to guarantee a solution for all tasks and the collision avoidance. The first subtasks are described as

$$\phi_{3,a} := \|p_1 - p_2\|_2 \leq d_{2,a} \tag{43}$$

and

$$\phi_{3,b} := \|p_1 - p_2\|_2 \leq d_{2,b} \tag{44}$$

where $d_{2,a}$ and $d_{2,b}$ are the maximum distances between the centers of the robots during formation. d_2, a was set to 1.3



(a) (b) (c) (d)

Figure 2. γ-functions for both cases, where $\gamma_{\text{D/E}}$ corresponds to γ_2, γ_{form} corresponds to γ_3, γ_{pre} corresponds to γ_1 and γ_{R} corresponds to γ_4. (2a) All γ-functions over time for case without consideration for collision avoidance. (2b) The same as 2a only without γ_{R}-function for easier comparison. (2c) All γ-functions over time for case with collision avoidance. (2d) The same as 2c only without γ_{R}-function for easier comparison.

meters, i.e. two times the agent's radius plus 1 meter. $d_{2,b}$ was set to 2.3 meters i.e. two times the agent's radius plus 2 meters. The predicate functions are defined as

$$h_{7,a}(\mathbf{x}) = 1.3 - \|p_1 - p_2\|_2 \tag{45}$$

and

$$h_{7,b}(\mathbf{x}) = 2.3 - \|p_1 - p_2\|_2. \tag{46}$$

The second subtask instructed agent 2 to reach the check-point area R, a circle with a radius of 5.15 meters and a center point in $[0; 0]$. The final task is described as

$$\phi_4 := \|p_2 - p_R\|_2 \leq d_1 \tag{47}$$

where as in the previous tasks, the largest distance between the agent and the check point area is $d_1 = 5$ meters. The predicate function is defined as

$$h_8(p_2) = d_1 - \|p_2 - p_R\|_2. \tag{48}$$

The formula for the whole STL task is defined as

$$\phi := \phi_{env} \wedge \phi_1 \wedge \phi_2 \wedge \phi_3 \mathcal{U}_{[15,25]} \phi_4. \tag{49}$$

D. Temporal Behaviour

For the predicate functions h_i, where $i = 1, ..., 4$ and 7, the linear function (17) for the temporal behaviour was used. The exponential function (18) together with (19) was used when $i = 5, 6, 8$ with the desired robustness $r = 0.01$.

The values for γ_∞, γ_0 and t^* differ depending on the task, giving γ separate values. γ_1 was calculated for the predicate described in equation (34) i.e. when $i = 1, ..., 4$ for h_i. For the tasks described in equations (38) and (40), when $i = 5$ and 6 respectively, γ_2 was calculated. Since both task 1 and 2 were similar for the two agents to reach their respective areas D and E, the tasks had the same γ. When $i = 7$ for the formation task in (43) and (44), γ_3 was used for the temporal behaviour. Finally, γ_4 was calculated for the final task described in (47), when $i = 8$, for agent 2 to reach area R.

t_j^* for the temporal behaviour is selected according to when the corresponding task should be satisfied, as defined in equation (20). $\gamma_{j,0}$ and $\gamma_{j,\infty}$ are selected by tuning, with the conditions that $\gamma_{j,0}$ can be infinitely small and $\gamma_{j,\infty}$ must be greater than or equal to the robustness value r.

Table I
TUNING VALUES FOR RESULT WITHOUT COLLISION AVOIDANCE

j	$\gamma_{j,\infty}$	$\gamma_{j,0}$	t_j^*
1	0.6	-20	0
2	0.05	-100	10
3	0.05	-100	15
4	0.5	-10 000	25

Table II
TUNING VALUES FOR RESULT WITH COLLISION AVOIDANCE

j	$\gamma_{j,}$	$\gamma_{j,0}$	t_j^*
1	0.6	-20	0
2	0.05	-100	10
3	0.05	-50	15
4	0.5	-8 000	25

The values used for each γ_j in each result is given in Table I for the case without consideration of collision avoidance, and in Table II for the case with consideration of collision avoidance. See Fig. 2a and 2b for γ with these values plotted over time for the case without collision avoidance. See Fig. 2c and 2d for γ with these values plotted over time for the case with collision avoidance.

E. Algorithms

Here follows descriptions of all algorithms coded by the authors for this project. Algorithm 1 describes the calculation process of the four predicate functions defined by (37). Algorithm 2 explains the calculations corresponding to both γ functions defined by (17) and (18). Algorithm 3 is for the main simulation program, calling both algorithm 1 and 2 while progressing through the simulations. Algorithm 3 runs until the ending time, set to 25 seconds. All algorithms have been coded in MATLAB and the results have been plotted in figures for a better visual understanding. The simulations were performed on an AMD Ryzen 7 3700U 2.3 GHz CPU with 8 GB of RAM. On average, solving the MATLAB `quadprog` function took 5.7 ms for the scenario where collision avoidance is not active and 5.2 ms for the scenario where collision avoidance is active.

Algorithm 1: Predicate function $h(\mathbf{x})$ with gradients algorithm

Result: $h_i(\mathbf{x})$ and corresponding gradients
$h_1 = \varepsilon - p_x$;
$\frac{dh_1}{d\mathbf{x}} = [-1 \quad 0]^T$;
$h_2 = \varepsilon + p_x$;
$\frac{dh_2}{d\mathbf{x}} = [1 \quad 0]^T$;
$h_3 = \varepsilon - p_y$;
$\frac{dh_3}{d\mathbf{x}} = [0 \quad -1]^T$;
$h_4 = \varepsilon + p_y$;
$\frac{dh_4}{d\mathbf{x}} = [0 \quad 1]^T$;

Algorithm 2: γ function with gradient algorithm

Result: $\gamma(t)$ and corresponding gradient
if *The γ function is linear* **then**
  if *time $t < t^*$* **then**
    $\gamma = \frac{\gamma_\infty - \gamma_0}{t*} t + \gamma_0$;
    $\frac{d\gamma}{dt} = \frac{\gamma_\infty - \gamma_0}{t*}$;
  else
    $\gamma = \gamma_\infty$;
    $\frac{d\gamma}{dt} = 0$;
  end
else if *The γ function is exponential* **then**
  $\gamma = (\gamma_0 - \gamma_\infty)e^{-\mathfrak{l}t} + \gamma_\infty$;
  $\frac{d\gamma}{dt} = -\mathfrak{l} \cdot (\gamma_0 - \gamma_\infty)e^{-\mathfrak{l}t}$;
end

Algorithm 3: Main simulation algorithm

Result: Updated state **x** for the agents
Define initial state for agents;
Define initial coordinates and goal area;
Define values for γ functions;
Define constants for collision avoidance;
while *time t < end time = 25 seconds* **do**
  Calculate γ functions and gradients for γ functions;
  Define tasks;
  Calculate predicate function, $h(\mathbf{x})$, and gradient for predicate function;
  Calculate $\mathfrak{b}_i(\mathbf{x}, t)$ and its gradients $\frac{\partial \mathfrak{b}_i(\mathbf{x},t)}{\partial \mathbf{x}}$ and $\frac{\partial \mathfrak{b}_i(\mathbf{x},t)}{\partial t}$ for tasks;
  if $t < t_i^*$ **then**
    $\mathfrak{o}_i = 1$
  else
    $\mathfrak{o}_i = 0$
  end
  Calculate $\mathfrak{b}(\mathbf{x}, t)$ and its gradients $\frac{\partial \mathfrak{b}(\mathbf{x},t)}{\partial \mathbf{x}}$ and $\frac{\partial \mathfrak{b}(\mathbf{x},t)}{\partial t}$;
  Calculate collision avoidance term;
  Define vector A and scalar b;
  Use `quadprog` to solve for **u**;
  Update state for agents;
  Update time;
end

V. Simulations

The following describes the mathematics used for the simulations which have been made in Matlab based on the three algorithms.

In these simulations, the collision avoidance is defined by (15) and (16) where $k = [0.45\ 0.9]$.

To find the control input **u**, the barrier for every task had to be separately calculated with their corresponding gamma, see equation (17) and (18),

$$\mathfrak{b}_i(\mathbf{x}, t) = \begin{cases} -\gamma_1(t) + h_i(\mathbf{x}) & \text{when } i = 1, ..., 4 \\ -\gamma_2(t) + h_i(\mathbf{x}) & \text{when } i = 5, 6 \\ -\gamma_3(t) + h_i(\mathbf{x}) & \text{when } i = 7 \\ -\gamma_4(t) + h_i(\mathbf{x}) & \text{when } i = 8 \end{cases}$$

where each i is explained in IV. *D. Temporal behaviour*. The total barrier for the system was then calculated as defined in equation (28).

See Fig. 3 for the total barrier plotted over time for the cases with and without consideration to collision avoidance.

In order for the tasks to be satisfied, the barrier needs to stay positive throughout the simulation. Fig. 3a and 3b display plots of the barrier over the entire simulation time for both results.

Figure 3a shows the barrier for the first result, which did not consider collision avoidance. The same is shown in figure 3b, where collision avoidance was implemented.

The following mathematical system was used in Matlab's function `quadprog`

$$\min_{\mathbf{u}} \frac{1}{2}\mathbf{u}^T H \mathbf{u} + f_u^T \mathbf{u} \tag{50}$$

and

$$A\mathbf{u} \leq b \tag{51}$$

where the matrix and the vectors were, in this project, chosen to

$$H = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}, \tag{52}$$

$$f_u = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T, \tag{53}$$

$$A = -\frac{\partial \mathfrak{b}}{\partial \mathbf{x}} = -\begin{bmatrix} \frac{\partial \mathfrak{b}}{\partial x_1} & \frac{\partial \mathfrak{b}}{\partial x_2} & \frac{\partial \mathfrak{b}}{\partial x_3} & \frac{\partial \mathfrak{b}}{\partial x_4} \end{bmatrix}, \tag{54}$$

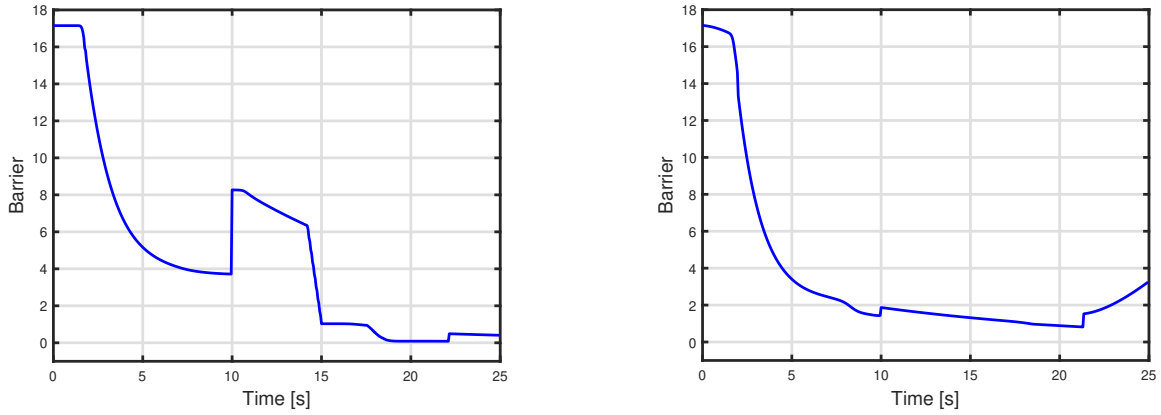
and

$$b = \alpha \cdot \mathfrak{b}(\mathbf{x}, t) + \frac{\partial \mathfrak{b}}{\partial t} + \frac{\partial \mathfrak{b}}{\partial \mathbf{x}} f(\mathbf{x}) \tag{55}$$

where $\mathfrak{b}$(**x**,t) is described in equation (28), $\frac{\partial \mathfrak{b}}{\partial t}$ is described in equation (33) and $\alpha = 0.07$.

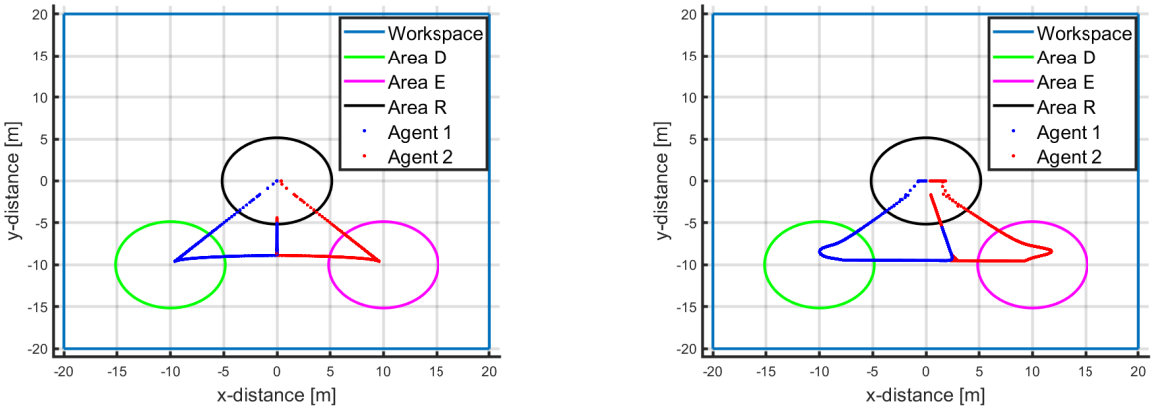
`Quadprog` returns

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}. \tag{56}$$



(a) (b)

Figure 3. Barrier functions plotted over time. (3a) Barrier for the case without consideration to collision avoidance. (3b) Barrier for the case with consideration to collision avoidance.



(a) (b)

Figure 4. The agents' positions plotted through both simulations. (4a) Agent 1 (left) and 2 (right) start in their initial positions in and close to the origin of the space. They move down diagonally to their separate areas D (left) and E (right), they then meet up to move in a formation back towards their starting area, where the formation task is discarded as soon as agent 2 has reached area R. No collision avoidance mechanism is applied. (4b) Agent 1 and 2 once again start in their initial positions in and close to the origin of the space. They move down to their separate areas D and E, they meet up, but before they move in a formation towards their starting area, agent 2 moves backwards at first to avoid collision with agent 1. The formation task is discarded as soon as agent 2 has reached area R. The collision avoidance mechanism is applied.

Then both agents' states were updated with a discrete time step $\Delta t = 0.05$ seconds and returned in a vector containing their new coordinates

$$p^{t+\Delta t} = p^t + \mathbf{u}^t \cdot \Delta t = \begin{bmatrix} p^t_{1,x} \\ p^t_{1,y} \\ p^t_{2,x} \\ p^t_{2,y} \end{bmatrix} + \begin{bmatrix} u^t_1 \\ u^t_2 \\ u^t_3 \\ u^t_4 \end{bmatrix} \Delta t = \begin{bmatrix} p^{t+\Delta t}_{1,x} \\ p^{t+\Delta t}_{1,y} \\ p^{t+\Delta t}_{2,x} \\ p^{t+\Delta t}_{2,y} \end{bmatrix}. \quad (57)$$

The agents updated their position and resolved the QP problem (30)-(31). This entire process was repeated until the set ending time, which was in both simulations set to 25 seconds.

The paths of both agents were plotted for the cases with and without consideration to collision avoidance and can be seen in Fig. 4.

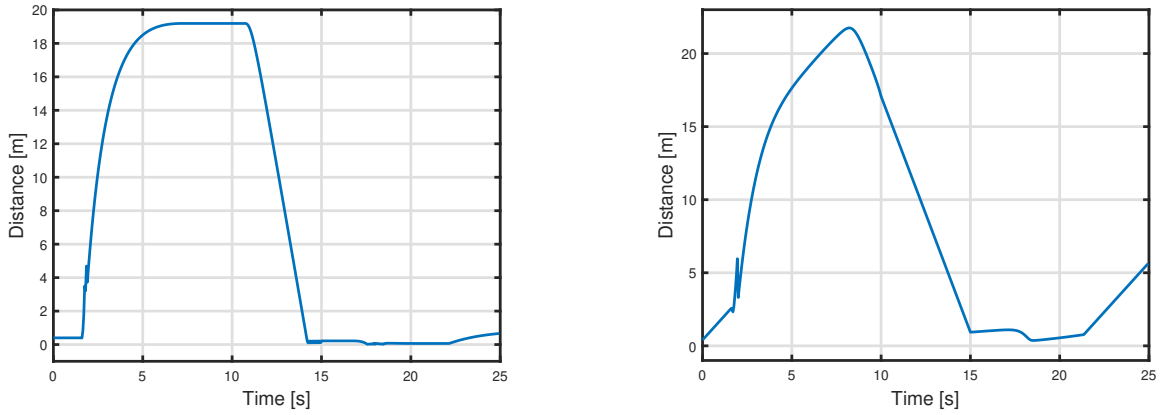
The distance between the two agents from center to center, $\|p_1 - p_2\|_2$, was measured during both simulations to check for collisions and can be seen in Fig. 5.

VI. Discussion

A. Comparison of Results

As seen in Fig. 4 the agents succeeded in satisfying the spatial tasks both with and without collision avoidance. However there are some differences between the two cases.

To ensure that the barrier function was positive at all times with the collision avoidance there was a need to increase the maximum allowed distance for the robots in the formation. Hence there is a difference in the definition of the two formation tasks (43) and (44) for each case. This was made to guarantee positive barrier for the collision avoidance scenario. In the end as seen in Fig. 3a and 3b both barrier functions remained non-negative throughout the whole simulation meaning that all tasks were satisfied. It would have been desirable



(a) (b)

Figure 5. The distance between the center points of the agents, $\|p_1 - p_2\|_2$, plotted over time. (5a) The distance is plotted as no collision avoidance mechanism is applied. The distance is close to 0 at some points. (5b) The distance is plotted as the collision avoidance mechanism is applied. The distance between the agents never reaches below 0.37 meters.

to be able to use the same definition for the formation task, i.e. to have the same maximum allowed distance between the agents, when considering the collision avoidance as for the one without. The goal for the formation task was for the agents to be able to maintain a fairly short distance between each other. For a scenario in a warehouse, it would have been preferable to guarantee a smaller maximum distance if the robots were to e.g. transport an item together.

B. Comments on Plots

The trajectory without consideration for collision avoidance in Fig. 4a is a lot more even and efficient than the case with collision avoidance in Fig. 4b. This is likely because of there being no conflicting tasks to satisfy, giving the controller a natural path to perform the tasks. The collision avoidance always poses as a conflicting task as soon as the agents approach or are close to each other, which made the paths of the agents more irregular and less natural.

The discontinuities seen in the plots of the barrier, Fig. (3a) and (3b), are due to the barrier only being piece wise differentiable. These occur because of the deactivation policy, $\mathtt{o_i}$, where some tasks become deactivated at certain points in time.

C. Collision Avoidance and Formation Control

The difficulty of implementing both a collision avoidance mechanism and a formation task is that they are contradicting. This means that the controller will have to consider one task defined to make the agents stay close to each other (formation control) and one assignment to keep the agents away from each other (collision avoidance). This gives the controller a limited space for satisfaction of both tasks, which is difficult to tune in.

Collision avoidance is desired in systems with several agents since it ensures safety for the agents. Fig. 5b shows that the distance between the agents' center points when collision avoidance was considered never reached below 0.37 meters. However for the distance shown in Fig. 5a without collision avoidance the minimum distance between the agents was close to 0. Since the agents have a radius of 0.15 meters at least 0.3 meters between the robots was necessary to prevent them from colliding. When applying these scenarios in reality, the collision avoidance is one of the most important assignments for the system. Even though the trajectory for the case with collision avoidance is not as straight forward as the other one, it is still preferable.

D. Future Work

Future work would lie within the interest to further explore the possibilities with STL defined tasks performed with a controller based on control barrier functions. A development of this project could be to make the simulation environment resemble a warehouse in even more detail, for example adding obstacles for the agents to avoid while navigating the area, solo or in i a formation. Another extension could be to add more robots to the formation. If a large piece has to be moved by more than one robot, the formation would likely require at least 3 to 4 robots in the formation for stability in the transportation of the object.

Further exploration in combining the collision avoidance and formation task would have been of great interest for a future project. Because of their contradicting nature there were some difficulties to make these segments co-operate in this project and a continued development to improve this would have been beneficial.

VII. Conclusion

The purpose of the study was to simulate a warehouse environment with STL tasks and perform them with a controller based on control barrier functions. Two agents were tasked to move from a starting point to two separate areas, and then rendezvous to move in a formation back towards their starting area. All this while staying within their designated workspace. The motion plan was created in two versions,

one with consideration to collision avoidance and one without this consideration. While the case without collision avoidance proves more even paths for the agents, the case with collision avoidance is still preferable, despite the more irregular paths. To apply collision avoidance to STL tasks simultaneously as a formation task is in effect, is an achievement and also a more realistic approach to the warehouse environment.

ACKNOWLEDGMENT

The authors would like to thank our supervisor Maria Charitidou for her amazing support and patience during this project. This project would not have been possible without her guidance.

REFERENCES

[1] D. Berreby. (2020, Aug.) The robot revolution has arrived. National Geographic. [Online]. Available: https://www.nationalgeographic.com/magazine/article/the-robot-revolution-has-arrived-feature

[2] E. Ackerman. (2020, Mar.) Autonomous robots are helping kill coronavirus in hospitals. We Care for Humanity. CA. [Online]. Available: http://www.wecareforhumanity.org/uploads/1/5/1/4/15147010/autonomous_robots_are_helping_kill_coronavirus_in_hospitals.pdf

[3] L. Lindemann and D. V. Dimarogonas, "Control barrier functions for multi-agent systems under conflicting local signal temporal logic tasks," *IEEE Control Systems Letters*, vol. 3, no. 3, pp. 757–762, 2019.

[4] The Lexico website. (2021, Apr.). [Online]. Available: https://www.lexico.com/definition/temporal

[5] P. Wieland and F. Allgöwer, "Constructive safety using control barrier functions," *IFAC Proceedings Volumes*, vol. 40, no. 12, pp. 462–467, 2007, 7th IFAC Symposium on Nonlinear Control Systems. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667016355690

[6] L. Lindemann and D. V. Dimarogonas, "Control barrier functions for signal temporal logic tasks," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 96–101, 2019.

[7] L. Lindemann, "Planning and control of multi-agent systems under signal temporal logic specifications," Ph.D. dissertation, KTH Royal Institute of Technology, Stockholm, 10 2020.

[8] T. Kopfstedt, M. Mukai, M. Fujita, and O. Sawodny, "Formation control for mobile robots in partially known environments using mixed integer programming and fuzzy systems," in *2006 SICE-ICASE International Joint Conference*, 2006, pp. 1832–1837.

[9] X. Zhou, Y. Zou, S. Li, and H. Fang, "Distributed receding horizon control for multi-agent systems with conflicting siganl temporal logic tasks," in *2020 2nd International Conference on Industrial Artificial Intelligence (IAI)*, 2020, pp. 1–6.

[10] V. Raman, A. Donzé, M. Maasoumy, R. M. Murray, A. Sangiovanni-Vincentelli, and S. A. Seshia, "Model predictive control with signal temporal logic specifications," in *53rd IEEE Conference on Decision and Control*, 2014, pp. 81–87.

[11] L. Lindemann and D. V. Dimarogonas, "Barrier function based collaborative control of multiple robots under signal temporal logic tasks," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 4, pp. 1916–1928, 2020.

[12] L. Lindemann and D. V. Dimarogonas, "Decentralized control barrier functions for coupled multi-agent systems under signal temporal logic tasks," in *2019 18th European Control Conference (ECC)*, 2019, pp. 89–94.

[13] L. Lindemann and D. V. Dimarogonas, "Robust control for signal temporal logic specifications using discrete average space robustness," *Automatica (Oxford)*, vol. 101, pp. 377–387, 2019.

CONTEXT C – PART I
LEARNING IN DYNAMICAL SYSTEMS

POPULAR DESCRIPTION

The "Foolish" Algorithm that Became Historic

Deep Blue beat Kasparov. Watson beat Jeopardy! AlphaGo beat Sedol. The story of the machine that beat the professional has over the last three decades been told many times. Artificial intelligence is not only catching up with human intelligence, but in some cases even surpassing it.

In the history of computers, engineers have been fascinated with the idea of training computers to play different types of games. In 2016, Lee Sedol who at the time was one of the best Go-players in the world, played a match against AlphaGo in which the AI made a move that surprised both commentators and professionals. The now famous move 37 was initially seen as a blunder but was later cited as the move that enabled AlphaGo to win the game.
AlphaGo claimed victory over Sedol implementing a machine learning technique referred to as reinforcement learning. All reinforcement learning algorithms have a simple goal - maximizing some reward over time. In the context of games, this reward is the score or amount of games won. But how do machines maximize these rewards? They practice, like humans do.

By letting AlphaGo play against itself, over the course of millions of games, it progressively learned the do's and don'ts of the game. In the end, AlphaGo managed to accumulate thousands of years of human knowledge during a period of just a few days and even awed the audience during the games by unconventional strategies and creative new moves.

We are already seeing these techniques applied to more tangible matters as the biological sciences. AlphaFold, a descendant of AlphaGo, are now channeling these superpowers to predict the three-dimensional structure of proteins by their aminoacid sequence. The method, if successful, may reduce experimentation costs in drug research and allow for more efficient drug discovery.

SUMMARY OF PROJECT RESULTS

An increasing number of problems in our technological world are hard to define and to solve in a traditional manner. The tasks may be predicting stock markets, design of collision avoidance systems or to exceed human performance in games. Pattern relying tasks like these have been hard to model with classical mathematical models and programming. New complex pattern recognition models, based on machine learning, are proving to be the key for solving these kinds of problems. The main idea behind the models is to use data from past events in order to accurately predict future events. The way the algorithms adapt in human-like ways to creatively solve various tasks are sparking excitement in the entire technology community.

The project groups in C1 have explored machine learning algorithms for predicting patterns in financial markets. Out of many applicable machine learning methods for predicting patterns in financial markets, the focus of the groups has been to examine the so-called Hidden Markov Models and Support Vector Regression as methods. Hidden Markov Models is a method where the process being modelled is assumed to be independent of the past and where the output are dependent on unobservable hidden states. The goal of Support Vector Regression is to find a hyperplane that fits data with the smallest possible error.

These methods have been implemented through training based on historical price data from financial markets to thereafter be able to make predictions. The aim of the two projects is to evaluate the algorithms that have been mentioned above, and their functionality and viability as tools in real-world applications.

There are a number of opportunities regarding future work - an algorithm able to model a system as dynamical as a financial market would be useful in other fields as well. Increased prediction accuracy would be a major focal point in the future, especially when the vast majority of algorithms decrease in accuracy when making predictions over longer time periods. For future projects, an area of interest would be to investigate how the used algorithms perform in fields similar to the financial markets.

The financial markets have always been a place of volatility, where information and the ability to extrapolate have been essential. These abilities have for the longest time been centralised within the financial institutions, but with machine learning becoming readily available, the playing field might be evening itself out. With the democratization of this technology, retail investors might be able to bridge the gap to financial institutions and invest with minimized risk and maximized possibility of return. Due to the financial incentives further research will and ought to be conducted in the future.

In project groups C2, reinforcement learning (RL) algorithms were implemented in game environments in order to find an optimal strategy to beat them. RL was used since the different states and dynamics of the environment are unknown from the beginning. The environments of the projects are taken from OpenAI, a development and research company with many implemented environments for games. The specific game environments that were used for the projects were CartPole, where the objective is to balance an inverted pendulum, and Flappy Bird, a popular mobile game where a bird needs to be flown between gaps. In these environments, the agent needs to explore the dynamics of the system in order to gain experience and develop the optimal policy. No model of the environment is provided to the agent, which means, the algorithms are model free. Algorithms that have been used in the projects are Q-learning, Deep Q-learning and A3C. The aim of the projects is to combat different difficulties the algorithms face in these environments, for example sparse rewards, which is when the environment has periods where exploring will not yield any additional useful information until a certain milestone is reached.

For future projects, further enhancements of the algorithms could be investigated, such as new methods for increasing exploration, which would broaden the spectrum of environments where the algorithms could be applied. It could also be of interest to investigate environments with larger action spaces.

In many real-world applications the knowledge of a system is limited and thus it is not possible to create a model in the traditional sense. RL provides a framework which makes it possible to make optimizations in these types of systems by mimicking human learning. Games provide environments that cannot be easily modeled, but they are relatively simple and controllable, which is why they serve as a good starting point when first getting in touch with RL and for developing the algorithms. It is easy to observe the effect of a change of an algorithm which makes these environments perfect for testing.
The project groups in C3 used RL algorithms to optimize the movements of several virtual self-driving robots within a confined warehouse environment. The goal was to enable these robots to navigate from their starting positions to their target positions while avoiding collisions.

To solve the problem, the groups considered multiple approaches. One of the methods used was Deep Q-learning, a well known method in the field of machine learning. A lesser established method named Multi-agent Rollout was also used in conjunction with neural networks.

Multiple paths can be taken for further development of the algorithms used. A possible development is to limit the information each robot has access to, to see if the robots can still accomplish their given task successfully. It could also be relevant to investigate how well the chosen methods works when applied to real world scenarios.
The purpose of implementing self-driving robots in a warehouse environment is to relieve humans of tasks that are monotonous and/or physically straining. These types of robots are becoming increasingly common in warehouses around the world, but they are still expensive in relation to their performance, which is why further research in this area is necessary.

In conclusion the work in the different projects contribute to the area of learning in dynamical systems, as different machine learning algorithms have been investigated. It is shown that these methods can be applied on a wide variety of complex problems, some previously unsolvable. Machine learning has a multitude of real-world applications which will impact us greatly and therefore it is of importance to discuss the impact on society and the environment.

IMPACT ON SOCIETY AND ENVIRONMENT

During the last decade, with growing datasets and ever increasing computer performance, humanity has gotten one of the most exciting tools since the invention of calculus – machine learning. The promises are endless with algorithms wielding superpowers never seen before and we have already seen proof of its subversive power when looking at the jumping and dancing robots created by Boston Dynamics. There are many utopian dreams about a future where robots are doing all the work and mankind reaping its benefits, but there are a few possible pitfalls related to the technology worth discussing – moral culpability, joblessness caused by automation, personal integrity and the environmental impact.

The increasing complexity of machine learning algorithms gives us the possibility to automate tasks before seen as distinctly human. The unanimous perception has always been that low-skilled work is the first to go, which is partially true, but with

the advent of machine learning it is more about the repetitiveness and predictability of both cognitive and physical tasks. Therefore, previously white-collar jobs such as physicians and accountants are seeing themselves threatened when the wave of automation comes rolling. Due to machine learning algorithms becoming increasingly complex and able to perform complex tasks, the ability of states to tackle possible future unemployment in the wake of automation becomes of utter importance.

On the other hand, major technological advancements have historically created more jobs and wealth than they have destroyed, and therefore they might instead transform the characteristics of future jobs – from repetitive task towards creative tasks, where humans excel and automation takes on a supportive role. So, instead of relieving mass unemployment it might be the purpose of the future state to make it possible for its citizens to learn new skills and keep up in an ever-changing job market. The increasing cognitive demand will most likely be felt asymmetric through the population where people not able to keep up will lag behind. This is where the cohesiveness and core values of the society will be on display, and of utmost importance.

But if machines are becoming autonomous, making decisions based solely on their soft- ware – who are morally culpable if something were to happen? Machines cannot be culpable in any satisfactory sense and therefore we will turn to the people behind the machine – the soft- ware engineers developing the algorithm, the company owning its intellectual property and the regulatory body which did not foresee this accident happening. The complexity of the matter might nip the technology in the bud, by outlawing autonomous machines entirely before even becoming viable.

Inadvertently, one inherent problem with machine learning is also its primary strength – its ability to reflect the data it is being fed. In the context of predicting closing stock prices it does not matter, but in the context where machines make decisions based on datasets skewed by, for example, structural inequalities it would exacerbate and reinforce that skewness, or inequality. This is something worth thinking about, especially when more and more tasks get digital and automated.

Then it is the issue of personal integrity. In a world with more connected devices than ever before, companies whose business model dabbles in relaying information about its users, machine learning is, and will, be a significant component of making it as profitable. With the increasing accuracy of machine learning and more personal information being available than ever before, there will come a time when our wants and needs are totally predictable - and then marketing companies will target you. Is that moral and good customer service or immoral and breaching personal integrity? It's a thin line and the subject is debatable. But what about algorithms denying you either financial loans or job opportunity due to analysis done on your personal information available through devices and social media. Some people would say it makes the process more efficient, other would say it is discriminatory - what kind of information is viable for future machine learning algorithms to act on and if done, will it be possible to get an adequate explanation and then an option to repeal?

Lastly there are the possible environmental impacts of machine learning. Machine lear- ning algorithms base their decisions on computation en masse, which needs energy. Its energy consumption would become a sizeable amount in a future where machine learning permeates almost every non-human decision. But it is not to be forgotten that machine learning might be able to thwart this surge by being fundamental to future smart grids, but to what degree the technology will be able to modulate its own environmental impact is unknown. Then there are the indirect impact of machine learning - with the onset self-driving vehicle becoming omnipresent, they will need to be powered. They are mostly powered by batteries which is a technology littered with sustainability issues.

In short, no technology comes for free.

Stock Price Prediction Using SVR with Stock Price, Macroeconomic and Microeconomic Data

Idil Korkmaz and Simon Sandberg

***Abstract*—A wide variety of machine learning algorithms have been used to predict stock prices. The aim of this project has been to implement a machine learning algorithm using support vector regression to predict the stock price of two well known companies—Apple and Microsoft—one day into the future using the current day's stock price, macroeconomic data and microeconomic data and to compare the prediction error with the different data inputs. The results show that the addition of macroeconomic and microeconomic data did not improve the prediction error. This suggests that the macroeconomic and microeconomic data used in this project does not contain additional information about future stock prices. The results also show that support vector regression performs worse than linear regression, however in this case no definite conclusion can be drawn since only one kernel and a handful of parameter values were considered when training and testing the algorithm. However, these results might also suggest that using the current day's data is not sufficient to be able to predict the non-linear relationships.**

***Sammanfattning*—Ett flertal maskininlärnings-algoritmer har använts för att förutspå aktiepriser. Målet med det här projektet har varit att implementera en maskininlärnings-algoritm som använder sig av support vector regression för att förutspå aktiepriset av två välkända företag—Apple och Microsoft—en dag in i framtiden genom att använda dagens aktiepris, makroekonomisk data och mikroekonomisk data samt att jämföra prediktionsfelet med dem olika indata. Resultaten indikerar att additionen av makroekonomisk och mikroekonomisk data inte förbättrade prediktionsfelet. Detta antyder att den makroekonomiska och mikroekonomiska data som användes i projektet inte innehåller någon ytterliggare information om framtida aktiepriser. Resultaten indikerade också att linjär regression presterar bättre än support vector regression, men i detta fallet kan ingen definitiv slutsats dras eftersom endast en kernel och ett par parameter-värden användes för att träna och testa algoritmen. Däremot kan dessa resultat också antyda att dagens data inte är tillräcklig för att kunna förutspå dem icke-linjära förhållandena.**

***Index Terms*—Support Vector Regression, Stock Market Prediction, Macroeconomic Data, Microeconomic Data, Financial Markets**

Supervisor: Robert Bereza

TRITA number: TRITA-EECS-EX-2021:144

I. Introduction

The financial markets are complex systems that can be modeled as dynamical systems with many different parameters. The nature of these markets is such that they are highly influenced by a large number of factors, including both microeconomic and macroeconomic factors, for example the revenue of a company and the Real GDP of the country in which the chosen companies reside. The financial markets are comprised of many different types of markets where a wide variety of instruments are traded, where a stock is one of the most popular types of investment instrument. A stock is defined as a share of a company that is publicly traded on a financial market [1]. The main aim for companies when listing their shares on a publicly traded stock exchange is to raise capital for their future projects. The stock market allows both private and institutional investors to invest in companies of their choice. The aim of trading on financial markets is usually to make a profit, regardless of the type of actor or time frame for the investment. Hence, finding ways to take advantage of opportunities to maximize profits is of high interest, where stock price prediction is highly relevant. Stock price prediction refers to the attempt of trying to predict the future value or price of a stock. With an accurate forecasting tool, the potential for financial profits in the stock market could be very high. With the interest in the financial markets rising in the past year due to the unusual market conditions [2], stock price prediction is a exceedingly interesting topic to investigate.

In previous studies, a wide variety of methods and algorithms have been utilized to predict stock prices. Some algorithms and methods used are deep learning [3], hidden markov models [4], as well as support vector regression. Studies using support vector regression have shown that the algorithm has predictive power over short time periods, and that it can be considered a good tool to predict stock price trends [5] [6].

Project Description

In this project, we have aimed to implement the machine learning algorithm using support vector regression (SVR) to predict the stock price of two different stocks of two large, well-known companies - Apple and Microsoft. In particular, the aim has been to predict the stock price one day into the future by using the current day's stock price, macroeconomic data, and microeconomic data and to compare the performance of SVR with the different data inputs. The method and data inputs used are described below in further detail.

II. Theory

A. SVR

To be able to make stock price predictions we want to find a function that maps the current day's data to the next day's stock price. To do this we use SVR. SVR is a method that attempts to find a hyperplane that minimizes the distance

between the points that represent the real stock prices and the hyperplane. Ideally we would like to find $\mathbf{w}$ and b in the hyperplane equation,

$$y(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) + b. \tag{1}$$

such that

$$t_n \leq y(\mathbf{x}_n) + \epsilon \tag{2}$$

$$t_n \geq y(\mathbf{x}_n) - \epsilon \tag{3}$$

where $\boldsymbol{\phi}$ is a function that transforms points, $y(\mathbf{x}_n)$ is the predicted stock price, t_n is the real stock price and ϵ is the error we tolerate without penalizing a point. This means that we would like to find a hyperplane which is no more than the distance ϵ away from all points, which in this case represent the real values of the stock prices. However, this is not always possible, and therefore we have to introduce slack variables $\xi_n \geq 0$ and $\hat{\xi}_n \geq 0$ which allow for some of the real values of the stock prices to lie greater than the distance ϵ from the hyperplane. The conditions in this case become:

$$t_n \leq y(\mathbf{x}_n) + \epsilon + \xi_n \tag{4}$$

$$t_n \geq y(\mathbf{x}_n) - \epsilon - \hat{\xi}_n. \tag{5}$$

We therefore want to find $\mathbf{w}$, b, ξ_n, and $\hat{\xi}_n$ such that we minimize the error function,

$$C \sum_{n=1}^{N} (\xi_n + \hat{\xi}_n) + \frac{1}{2}\|\mathbf{w}\|^2 \tag{6}$$

subject to the constraints $\xi_n \geq 0$ and $\hat{\xi}_n \geq 0$ and the conditions in equation (4) and (5). This is a quadratic optimization problem where C is a regularization parameter and $\frac{1}{2}\|\mathbf{w}\|^2$ is a regularization term used to avoid overfitting. Overfitting means that the algorithm is overly specialized for the training data and therefore performs poorly when making predictions using data it has not encountered. It is conventional to use such a term, but we will not go into detail about it's exact effects here. We want to express this quadratic optimization problem in another form which gives us the ability to use a kernel function. To achieve this we introduce Lagrange multipliers and the so-called KKT conditions. These are common methods within optimization theory, but covering them is beyond the scope of this report. See [7, pp. 215–249] for a quick review of Lagrange multipliers. The following derivation is based on the derivation from [8]. Introducing the Lagrange multipliers $a_n \geq 0$, $\hat{a}_n \geq 0$, $\mu_n \geq 0$ and $\hat{\mu}_n \geq 0$ our Lagrangian becomes,

$$\begin{aligned}\tilde{L} = C \sum_{n=1}^{N} (\xi_n + \hat{\xi}_n) + \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} (\mu_n \xi_n + \hat{\mu}_n \hat{\xi}_n) \\ - \sum_{n=1}^{N} (a_n(\epsilon + \xi_n + y_n - t_n) + \hat{a}_n(\epsilon + \hat{\xi}_n - y_n + t_n)).\end{aligned} \tag{7}$$

We can then express this quadratic optimization problem in another form by using equation (1) and differentiating the Lagrangian with respect to $\mathbf{w}$, b, ξ_n and $\hat{\xi}_n$ and putting it equal to zero. This gives us the dual problem formulation which involves maximizing,

$$\begin{aligned}L = -\frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} (a_n - \hat{a}_n)(a_m - \hat{a}_m) k(\mathbf{x}_n, \mathbf{x}_m) \\ -\epsilon \sum_{n=1}^{N} (a_n + \hat{a}_n) + \sum_{n=1}^{N} (a_n - \hat{a}_n) t_n\end{aligned} \tag{8}$$

with respect to a_n and $\hat{a}_n$ and where we have introduced the so- called kernel $k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^T \boldsymbol{\phi}(\mathbf{x}')$. The kernel we use is the Radial Basis Function kernel which finds a hyperplane in infinite dimensions and in essence calculates how close two points $\mathbf{x}$ and $\mathbf{x}'$ are to each other and can be formulated in the following way

$$k(\mathbf{x}, \mathbf{x}') = exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2) \tag{9}$$

where γ is a parameter which determines how far the influence of a point reaches [9]. The kernel function enables for the computations to be made in the data space instead of having to transform the data into a higher dimensional feature space which can be computationally expensive. The RBF-kernel calculates the same value in the data space as the dot product of two points that has been transformed to infinite dimensional points with a simple exponential as seen in equation (9). This is often referred to as the kernel trick [10]. The RBF-kernel corresponds to a infinite-dimensional $\boldsymbol{\phi}$ and when you transform the points to infinite dimensions it becomes easier to find a infinite-dimensional hyperplane that lies close to the points. This infinite-dimensional hyperplane corresponds to a non-linear function in our data space. The constraints and conditions in this case becomes

$$0 \leq a_n \leq C \tag{10}$$

$$0 \leq \hat{a}_n \leq C \tag{11}$$

$$\sum_{n=1}^{N} (a_n - \hat{a}_n) = 0. \tag{12}$$

The optimization problem is to maximize the Lagrangian in equation (8) subject to the constraints and conditions in equation (10), (11) and (12). While differentiating equation (7) with respect to $\mathbf{w}$ and putting it equal to zero we obtained the relation

$$\mathbf{w} = \sum_{n=1}^{N} (a_n - \hat{a}_n) \boldsymbol{\phi}(\mathbf{x}_n) \tag{13}$$

If we insert this relation into equation (1) we obtain

$$y(\mathbf{x}) = \sum_{n=1}^{N} (a_n - \hat{a}_n) k(\mathbf{x}, \mathbf{x}_n) + b \tag{14}$$

which gives the function that can be used to predict the next day's stock price using the current day's data, where b is given by the following equation

$$b = t_n - \epsilon - \sum_{m=1}^{N} (a_m - \hat{a}_m) k(\mathbf{x}_n, \mathbf{x}_m). \tag{15}$$

III. Method

We use an algorithm using SVR with an RBF-kernel to predict the stock price of the Apple and Microsoft stock one day into the future by using the current day's stock price, macroeconomic data, and microeconomic data. In the following subsections we describe the data we use, the data processing and implementation of the algorithm, the evaluation metrics, and the training and testing of it.

A. Data

The data we use consists of 2516 data points between the time period 2005-01-03 and 2014-12-31 and every data point contains the stock price, six different macroeconomic numbers, two microeconomic numbers and the date if we use all the data. Macroeconomic and microeconomic data are presented monthly or quarterly, and therefore the data point for each month respectively each quarter is used on all the days where we have a stock price corresponding to that month or quarter.

A data point (X,Y) is defined in the following way if we use all the data,

$$X = \begin{bmatrix} Date_1 \\ Stockprice(Date_1) \\ Macro_1(Date_1) \\ Macro_2(Date_1) \\ Macro_3(Date_1) \\ Macro_4(Date_1) \\ Macro_5(Date_1) \\ Macro_6(Date_1) \\ Micro_1(Date_1) \\ Micro_2(Date_1) \end{bmatrix}, \quad Y = \left[Stockprice(Date_2)\right].$$

The data points corresponding to using only macroeconomic data and no macroeconomic or microeconomic data are found by removing the microeconomic numbers, respectively the macroeconomic and microeconomic numbers from the data points above.

1) Stock prices: The stock prices consist of the adjusted close price of the Apple and Microsoft stock and was gathered from Yahoo Finance [11]. The adjusted close price of a stock is the closing price amended to reflect any value change in the stock due to corporate actions. Traditionally, the adjusted close price is what is used for when doing analysis using historic stock price data [12].

2) Macroeconomic data: The macroeconomic data consists of the adjusted close price of the SP500 index and Dow Jones Industrial Average index [11], US Real GDP [13], US Unemployment Rate [14], the US M2 Money Supply [15] and the Consumer Price Index for All Urban Consumers: All Items in U.S. City Average [16].

The SP500 index is a market-capitalization-weighted index of the 500 largest publicly-traded companies in the US and the Dow Jones Industrial Average index tracks 30 large companies in the US that are not included in the SP500 index [17] [18]. US Real GDP refers to the real gross domestic product in the USA, which is the value of all goods and services produced in the United States, adjusted for inflation, measured in the local currency, USD [13]. The US Unemployment rate constitutes the percentage of unemployed people who are part of the labor force in the United States [14]. The US M2 Money Supply consists of M1 Money Stock plus savings deposits, individual retirement accounts, and balances in retail money market funds in the US [15]. The Consumer Price Index for All Urban Consumers: All Items in the U.S. City Average refers to the average monthly change in the prices of goods and services which are consumed by a majority of the population. This index indicates the price change from one time period to the next and is usually used as an indicator of inflation or deflation in a country's economy [16]. Together these macroeconomic indicators give one an overall view of how well the US economy is doing.

3) Microeconomic data: The microeconomic data consists of the revenue and net income for the Apple and Microsoft stock [19] [20]. The revenue of a company is a measure of the total sales under a period while the net income is the sales minus expenses, interest, and taxes [21] [22]. Together these two numbers gives one a good view of how well a company is doing.

B. Data Processing and Implementation

Firstly, since the data is in the form of a time-series the dates of every data point were converted to a numerical value. This is done by using the library datetime and the method $toordinal()$ which maps the date YYYY-MM-DD to the number of days from the date 01/01/01 to YYYY-MM-DD [23]. Secondly, the data used was in different order of magnitude and therefore scaling of the data was used to ensure that we minimize rounding errors. The scaling of the data is done by using sklearns $sklearn.preprocessing.StandardScaler()$ which transforms the data such that the data distribution will have mean 0 and variance 1 [24]. If we for example had a matrix with data where every column consisted of measurements of different variables, the values of all columns would be transformed such that every column has the distribution mean 0 and variance 1.

The algorithm using SVR is implemented in python where the library sklearn is used [25]. The class $sklearn.svm.SVR()$ is used to create an object with SVR and our chosen kernel and parameter values, and is trained and tested by using the methods fit and predict. The parameter values are the C-value, which is the regularization parameter, the γ-value, which determines how far the influence of a point reaches, and the ϵ-value, which specifies the ϵ-tube within which no penalty is associated in the error function with points predicted within a distance ϵ from the actual value [26]. Linear regression [27] has also been implemented using sklearns library which is used as a baseline to be compared with SVR.

C. Evaluation

To be able to measure the performance of SVR with the different data inputs one need to choose evaluations metrics. In this project, Root Mean Squared Error (RMSE) and the Mean Absolute Percentage Error (MAPE) are used as evaluation metrics. RMSE and MAPE are used because they are standard ways of measuring the error of predictions. The RMSE is defined as

$$RMSE = \sqrt{\sum_{n=1}^{N} \frac{((Pred\,val)_n - (Real\,val)_n)^2}{N}} \quad (16)$$

and the Mean Absolute Percentage Error (MAPE) is defined as

$$MAPE = \frac{100}{N} \sum_{n=1}^{N} \left| \frac{(Pred\,val)_n - (Real\,val)_n}{(Real\,val)_n} \right|. \quad (17)$$

where $Pred\ val$ is the predicted stock price and $Real\ val$ is the real stock price.

D. Training and Testing

When training and testing the algorithm one first wants to find the most optimal set of parameter values for the algorithm on a set of data points. In this case the parameter values are C, γ and ϵ. Once the most optimal parameter values have been found, the next step is to test the performance of the algorithm with this set of parameter values on another set of data points. Since we have a time-series data set, sklearn's time-series cross validation [28] was used to do this. The reason that time-series cross validation was used is because we want the evaluation process to look like the circumstances under on which the algorithm will be used. The time-series cross validation works in the following way.

Algorithm 1 Algorithm for training and testing

```
procedure TIME-SERIES CV
    n_0 = Initial number of training data points.
    n = n_0.
    N = The number of points you want to predict.
    while n ≠ n_0 + N do
        Train on data points:
        (X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n)
        Use X_{n+1} to predict Ŷ_{n+1}
        n = n + 1
```

Testing every possible value of C, γ and ϵ is computationally expensive, and therefore the following C, γ and ϵ values were considered, C = [0.2, 0.5, 1, 10, 100, 1000, 10000], γ = [0.1, 0.5, 1, 5, 10] and ϵ = 0.1. For the data consisting of 2516 data points, 2217 data points were used to find the parameter values giving us the lowest RMSE. The initial number of training data points was set to 1767 data points. This will give us 450 predicted values which are used to calculate the RMSE. This procedure was used on every combination of parameter values listed above.

Once the parameter values which result in the lowest RMSE have been found, the same time-series cross validation method used earlier is applied again on all the 2516 data points, where the initial number of training data points was set to 2217 data points and therefore we will get 299 predicted values which are used to calculate the RMSE and MAPE. This was done with the parameter values found earlier. The same training and testing method is used for linear regression with the exception that finding the most optimal parameters is not necessary.

IV. Results

In this section, we present the results of the training and testing done with our algorithm. As mentioned in the previous section, the first aim is to find the most optimal set of parameter values for the algorithm. These parameter values for the Apple and Microsoft stock can be seen below in Table I and II. The second aim was to test the performance of our algorithm with the parameter values found. The results can be seen graphically in Fig. 1-6, and numerically in Table III and IV.

TABLE I
C, γ AND ϵ FOR THE APPLE STOCK.

Apple: Method - Data	C	γ	ϵ
SVR: RBF-kernel - No macro or microdata	100	0.1	0.1
SVR: RBF-kernel - Macrodata	10	0.1	0.1
SVR: RBF-kernel - Macro and microdata	10	0.1	0.1

TABLE II
C, γ AND ϵ FOR THE MICROSOFT STOCK.

Microsoft: Method - Data	C	γ	ϵ
SVR: RBF-kernel - No macro or microdata	100	0.1	0.1
SVR: RBF-kernel - Macrodata	10	0.1	0.1
SVR: RBF-kernel - Macro and microdata	10	0.1	0.1

A. No Macroeconomic or Microeconomic Data

The predicted values of SVR and linear regression using no macroeconomic or microeconomic data is compared with the real values of both stocks in Fig. 1-2.

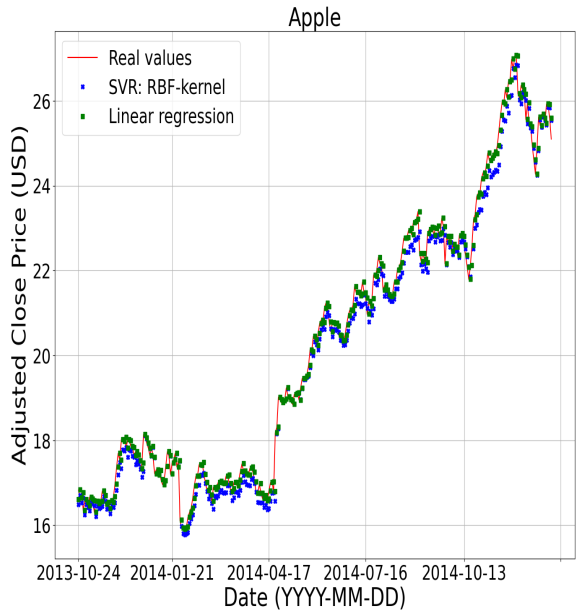


Fig. 1. Predicted adjusted close price for the Apple stock with SVR using RBF-kernel and linear regression with no macroeconomic or microeconomic data.

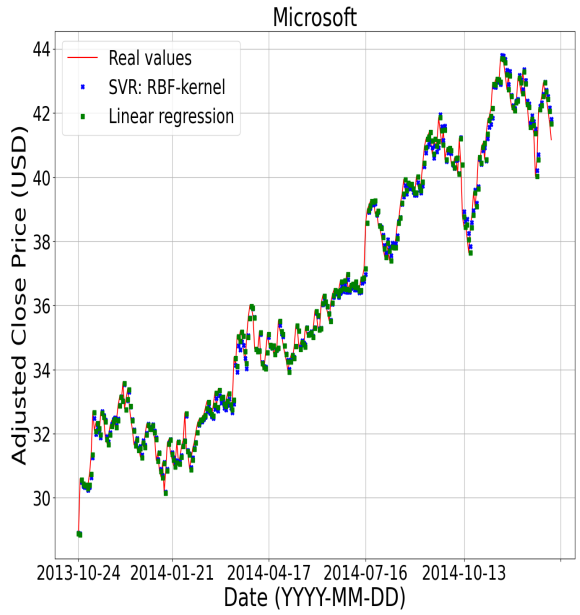


Fig. 2. Predicted adjusted close price for the Microsoft stock with SVR using RBF-kernel and linear regression with no macroeconomic or microeconomic data.

B. Macroeconomic Data

The predicted values of SVR and linear regression using macroeconomic data is compared with the real values of both stocks in Fig. 3-4.

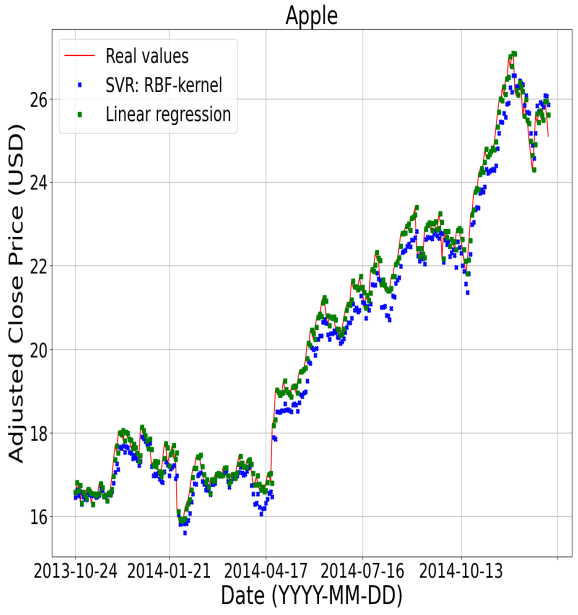


Fig. 3. Predicted adjusted close price for the Apple stock with SVR using RBF-kernel and linear regression with macroeconomic data.

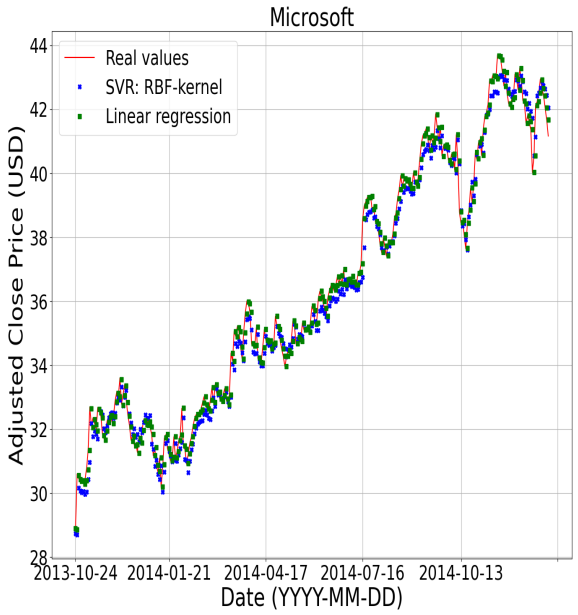


Fig. 4. Predicted adjusted close price for the Microsoft stock with SVR using RBF-kernel and linear regression with macroeconomic data.

C. Macroeconomic and Microeconomic Data

The predicted values of SVR and linear regression using macroeconomic and microeconomic data is compared with the real values of both stocks in Fig. 5-6.

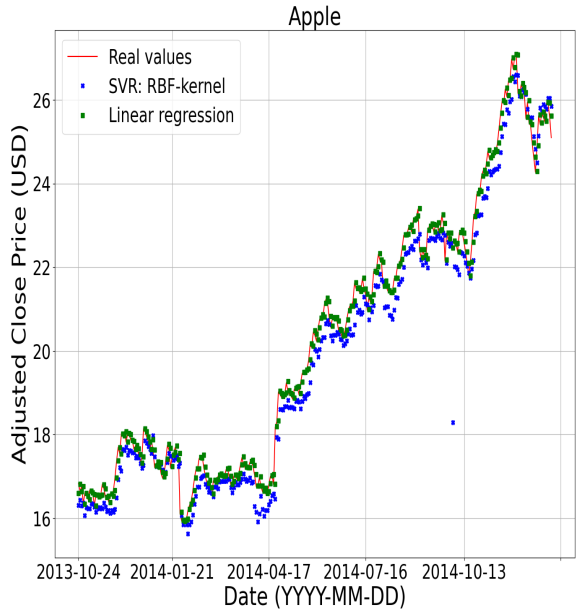


Fig. 5. Predicted adjusted close price for the Apple stock with SVR using RBF-kernel and linear regression with macroeconomic and microeconomic data.

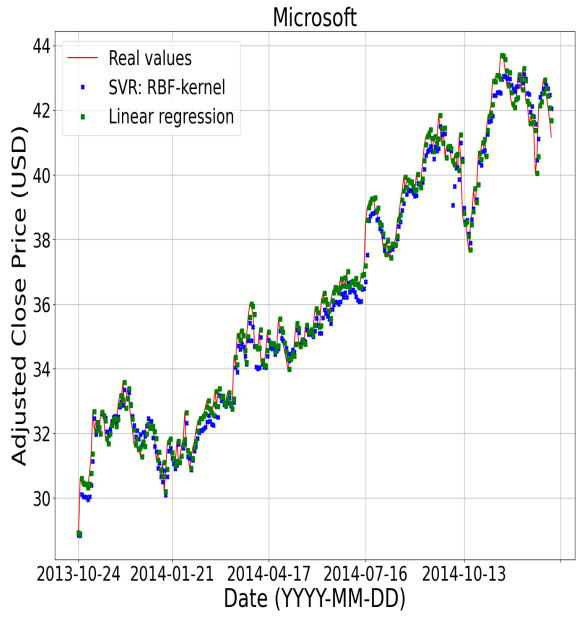


Fig. 6. Predicted adjusted close price for the Microsoft stock with SVR using RBF-kernel and linear regression with macroeconomic and microeconomic data.

D. All Data Inputs

In Table III and IV the RMSE and MAPE of SVR and linear regression with all the different data inputs is displayed.

TABLE III
RMSE AND MAPE FOR THE APPLE STOCK.

Apple: Method - Data	RMSE (USD)	MAPE (%)
SVR: RBF-kernel - No macro or microdata	0.3535	1.3373
SVR: RBF-kernel - Macrodata	0.4643	1.8749
SVR: RBF-kernel - Macro and microdata	0.5433	2.0811
Linear regression - No macro or microdata	0.2709	0.9552
Linear regression - Macrodata	0.2711	0.9559
Linear regression - Macro and microdata	0.2711	0.9557

TABLE IV
RMSE AND MAPE FOR THE MICROSOFT STOCK.

Microsoft: Method - Data	RMSE (USD)	MAPE (%)
SVR: RBF-kernel - No macro or microdata	0.4622	0.9534
SVR: RBF-kernel - Macrodata	0.5073	1.0791
SVR: RBF-kernel - Macro and microdata	0.5305	1.1319
Linear regression - No macro or microdata	0.4465	0.9359
Linear regression - Macrodata	0.4434	0.9288
Linear regression - Macro and microdata	0.4428	0.9306

V. DISCUSSION

In this section, we discuss the results of the training and testing done with our algorithm as well as future work that can be done. In Fig. 1-6 we can see the predicted stock price using SVR and linear regression compared with the real values. Both methods produce predictions that are close to the real values, which the MAPE confirms in Table III and IV.

We can also see in Table III and IV that the RMSE and MAPE increased when we used SVR with an RBF-kernel for both stocks when we added macroeconomic and microeconomic data. Therefore the addition of the macroeconomic and microeconomic data used did not improve the prediction error. This result is surprising since adding macroeconomic and microeconomic data provides information about how well the economy and the companies

are doing, which intuitively should give us information about future stock prices. A potential explanation for this is that SVR tries to find patterns in data, even if macroeconomic and microeconomic data is shown to be irrelevant. Finding patterns in irrelevant data is done at the cost of finding patterns in the data that is actually relevant. The results also show that linear regression performed better than SVR with all the different data input combinations. One possible explanation for why linear regression showed better performance is that only 35 combinations of parameter values were considered when finding the most optimal parameter values. Therefore it is possible that there are combinations of parameter values giving us an even lower RMSE and MAPE, even lower than the RMSE and MAPE gotten using linear regression. Another possible explanation is that only one kernel was considered, and therefore it might be the case that some other kernel performs better on this particular data set.

By looking at Fig. 1-6, both methods perform poorly when predicting stock price when there are major changes in trend or trend breaks. Both methods almost always predict that the trend will keep going and it's first after they have seen the new data that they realize that the price evolution has changed direction. However, linear regression seems to perform better than SVR between the trend breaks. In other words, a linear function seems to perform better than a non-linear function produced by SVR between the trend breaks. This might be because the price evolution between the trends breaks seems to be somewhat linear and a linear model will in most cases produce a better linear function than a non-linear model. This might be a reason that linear regression produces a smaller error than SVR. Something that is surprising is that the non-linear function produced by SVR doesn't capture the trend breaks changes very well. A possible reason why this is the case is that the current day's data might not be enough to be able to predict trend breaks. Therefore, it might be the case that to be able to predict trend breaks, you have to use data from longer time periods. Therefore SVR would probably have performed better if it was given more data to be able to predict trend breaks because linear regression would not have been able to describe trend breaks that is probably given by non-linear relationships.

An issue with the method used was that we have to retrain our algorithm on all the previous days every time we want to use it to predict one day into the future, making it computationally expensive to use. Hence, a consequence of this approach would be that if we for example took a data point 100 days into the future and try to predict the stock price the next day, our method would most likely produce a poor prediction since all the information between now and 100 days into the future would not have been used.

Future Work

As mentioned above, the testing of more combinations of parameter values and other kernels could be something that could be looked at to improve the prediction error even though this might be computationally expensive. In this project, we in essence found a function that maps the current day's data to the stock price tomorrow. It would be interesting to look at using data from the x previous days to predict the stock price for the next day and compare it with the results produced by our method. This method would probably produce better results since it would use the information from the x previous days compared to only using the current day's information to predict the stock price. It would therefore also probably perform better if you are trying to use it further into the future and also be able to predict trend breaks. Another possible future expansion that would be interesting to look at would be using other stocks or financial instruments, or training and testing the algorithm on a different time period to see how well it would perform on a wider variety of stocks or on a wider or narrower time period. It could be of interest to investigate the performance of SVR on more volatile stocks and volatile periods of time in the financial markets, such as stock market crashes or volatile activity in a specific financial instrument.

VI. Conclusion

In this project we have implemented the machine learning algorithm using SVR with an RBF-kernel to predict the stock price one day into the future by using the current day's stock price, macroeconomic data, and microeconomic data. The algorithm was used to predict 299 stock prices with different data inputs. The results show that the addition of macroeconomic and microeconomic data did not improve the prediction error. This suggests that the macroeconomic and microeconomic data used in this project does not contain additional information about future stock prices. The results also show that simple linear regression performs better than SVR with all the different data inputs, but in this case we can not draw any definite conclusion that SVR performs worse since only one kernel and 35 combinations of parameter values were considered when training and testing our algorithm. However, these results might also suggest that using the current day's data is not sufficient to be able to predict the non-linear relationships.

Acknowledgments

We would like to thank our supervisor Robert Bereza for helping us with this project. Without his guidance and support, this project would not have been possible.

References

[1] J. Berk, *Corporate finance.*, 3rd ed. Essex, UK: Pearson Education, 2013, ch. 1.3.

[2] Benjamin Curry, Anna-Louise Jackson. (2020, Dec.) 2020 stock market in review: A year that defied expectations. USA. [Online]. Available: https://www.forbes.com/advisor/investing/stock-market-year-in-review-2020/

[3] R. Singh and S. Srivastava, "Stock prediction using deep learning," *Multimedia tools and applications*, vol. 76, no. 18, pp. 18 569–18 584, 2017.

[4] A. Gupta and B. Dhingra, "Stock market prediction using hidden markov models," in *2012 Students Conference on Engineering and Systems*. IEEE, 2012, pp. 1–4.

[5] B. M. Henrique, V. A. Sobreiro, and H. Kimura, "Stock price prediction using support vector regression on daily and up to the minute prices," *The Journal of finance and data science*, vol. 4, no. 3, pp. 183–201, 2018.
[6] Y. Xia, Y. Liu, and Z. Chen, "Support vector regression for prediction of stock trend," in *2013 6th International Conference on Information Management, Innovation Management and Industrial Engineering*, vol. 2, 2013, pp. 123–126.
[7] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge: Cambridge University Press, 2004.
[8] C. M. Bishop, *Pattern recognition and machine learning.*, 1st ed. New York City: Springer, 2006, ch. 7.
[9] Sushanth Sreenivasa. (2020, Oct.) Radial basis function (rbf) kernel: The go-to kernel. Canada. [Online]. Available: https://towardsdatascience.com/radial-basis-function-rbf-kernel-the-go-to-kernel-acf0d22c798a
[10] Grace Zhang. (2018, Nov.) What is the kernel trick? why is it important? San Francisco, CA, USA. [Online]. Available: https://medium.com/@zxr.nju/what-is-the-kernel-trick-why-is-it-important-98a98db0961d
[11] (2021, Mar.) Yahoo finance. Yahoo, New York City, NY, USA. [Online]. Available: https://finance.yahoo.com/
[12] Akilesh Ganti. (2020, Dec.) Adjusted closing price. DotDash, New York City, NY, USA. [Online]. Available: https://www.investopedia.com/terms/a/adjusted_closing_price.asp
[13] U.S. Bureau of Economic Analysis. (2021, Mar.) Real gross domestic product. Saint Louis, Missouri, USA. [Online]. Available: https://fred.stlouisfed.org/series/GDPC1
[14] U.S. Bureau of Labor Statistics. (2021, Mar.) Unemployment rate. Saint Louis, Missouri, USA. [Online]. Available: https://fred.stlouisfed.org/series/UNRATE
[15] Board of Governors of the Federal Reserve System (US). (2021, Mar.) M2 money stock. Saint Louis, Missouri, USA. [Online]. Available: https://fred.stlouisfed.org/series/M2SL
[16] U.S. Bureau of Labor Statistics. (2021, Mar.) Consumer price index for all urban consumers: All items in u.s. city average. Saint Louis, Missouri, USA. [Online]. Available: https://fred.stlouisfed.org/series/CPIAUCSL
[17] Will Kenton. (2021, Mar.) S&p 500 index – standard & poor's 500 index. DotDash, New York City, NY, USA. [Online]. Available: https://www.investopedia.com/terms/s/sp500.asp
[18] Akilesh Ganti. (2021, Mar.) Dow jones industrial average (djia). DotDash, New York City, NY, USA. [Online]. Available: https://www.investopedia.com/terms/d/djia.asp
[19] (2021, Mar.) Apple income statement 2005-2021 — aapl. [Online]. Available: https://www.macrotrends.net/stocks/charts/AAPL/apple/income-statement?freq=%
[20] (2021, Mar.) Microsoft income statement 2005-2020 — msft. [Online]. Available: https://www.macrotrends.net/stocks/charts/MSFT/microsoft/income-statement?freq=Q
[21] Adam Hayes. (2021, Mar.) Revenue. DotDash, New York City, NY, USA. [Online]. Available: https://www.investopedia.com/terms/r/revenue.asp
[22] Will Kenton. (2021, Mar.) Net income (ni). DotDash, New York City, NY, USA. [Online]. Available: https://www.investopedia.com/terms/n/netincome.asp
[23] (2021, Apr.) datetime — basic date and time types. [Online]. Available: https://docs.python.org/3/library/datetime.html
[24] (2021, Apr.) sklearn standardscaler. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html#sklearn.preprocessing.StandardScaler
[25] (2021, Apr.) scikit-learn machine learning in python. [Online]. Available: https://scikit-learn.org/stable/
[26] (2021, Apr.) Sklearn svr. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html
[27] (2021, Apr.) sklearn linearregression. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
[28] (2021, Apr.) Sklearn timeseriessplit. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.TimeSeriesSplit.html

Using a Hidden Markov Model as a Financial Advisor

Emil Lindqvist and Robert Andersson

***Abstract*—People have been trying to predict the stock market since its inception and financial investors have made it their profession. What makes predicting the stock market such a hard task is its seemingly random dependency on everything from Elon Musks tweets to future earnings. Machine learning handles this apparent randomness with ease and we will try it out by implementing a Hidden Markov Model. We will model two different stocks, Tesla, Inc. and Coca-Cola Company, and try using the forecasted prices as a template for a simple trading algorithm. We used an approach of calculating the log-likelihood of preceding observations and correlated it with the log-likelihood of all the preceding subsequences of equivalent size by turning the time window by one day in the past. The results show that modeling two stocks of different volatility is possible, but using the result as a template for trading came back inconclusive with less than 50 percent successful trades for both of the modelled stocks.**

***Sammanfattning*—Människor har försökt förutsäga aktiemarknaden sedan starten och finansiella investerare har gjort det till sitt yrke. Det som gör att förutsäga aktiemarknaden till en så svår uppgift är dess till synes slumpmässiga beroende av allt från Elon Musks tweets till framtida intäkter. Maskininlärning hanterar denna uppenbara slumpmässighet med lätthet och vi kommer att testa det genom att implementera en Hidden Markov-modell. Vi kommer att modellera två olika aktier, Tesla, Inc. och Coca-Cola Company, och försöka använda de prognostiserade priserna som bas för en enkel algoritm att handla på. Vi använde ett tillvägagångssätt för att beräkna log-sannolikheten för föregående observationer och korrelerade den med log-sannolikheten för alla föregående följder av motsvarande storlek genom att vrida tidsfönstret med en dag tidigare. Resultaten visar att det är möjligt att modellera två aktier med olika volatilitet, men att använda resultatet som en mall för handel kom tillbaka oavgörande med mindre än 50 procent framgångsrika affärer för båda modellerna.**

***Index Terms*—hidden markov models, stock market prediction**

Supervisors: Yu Wang and Cristian Rojas

TRITA number: TRITA-EECS-EX-2021:145

I. Introduction

The stock market is a system that administers a platform for all large-scale economic transactions in the world at a dynamic rate labeled as stock value. Forecasting the stock value can potentially grant one with enormous profit opportunities, which have been a huge motivation for research in this area, both in academia and industry. The main hypothesis is that a probabilistically correct forecast can be extremely profitable, despite the problems like, dependence on time, seasonality and volatility. In detail, the tendency of stock market index prices point to the movement of the price index or the direction of fluctuation in the stock market index in the future. To make the correct financial decisions, the prediction of price trends is an essential tool. In spite of this essence, due to uncertainties and nonlinear factors convoluted in the data, prediction of financial time series is a tough task. In reality, a stock market is an extremely complex system, where the components who form the system, have changes in their prices without having significant patterns. Furthermore, the mood of the stock market is built upon various qualitative factors, such as natural, political and economic, which indicates non linearity and an enormous complexity to dimensionality [1]. However, investigating market behavior can lead to a better understanding of how moods alternate and thereby increases the chance of making profitable investment decisions. In general, one should seek to invest at the beginning of upward trends, namely termed as bullish markets and repel shares just in time before the prices fall again, namely termed as bearish markets.

The past years, a considerable amount of machine learning methods have been applied to the areas of financial time series prediction. There are numerous forecasting models of financial time series applying machine learning tools such as Support Vector Machines [2], Neural Networks [3] , Hidden Markov Models (HMM). HMM is a suitable approach when it comes to modeling sequential data, such as time series, based on the hypothesis of first-order Markov chain. As a matter of fact, due to the short-term and long term correlations found in empirical time series, Markov property plays an important role in financial time forecasting. In recent years, $HMMs$ have come into view as prominent tools for modeling financial time series. Hassan and Nath [4] developed an HMM for stock market forecasting, by trying to find some day in the past which is most alike with the current day in order to forecast the next day's stock price. Park and Lee [5] used continuous first-order HMM to forecast change in direction of next day's closing price. Nguyen [6] used $HMMs$ to forecast monthly closing prices and as consequence to derive an optimal trading strategy, which showed to exceed the conventional buy-and-hold strategy. All these applications establish that $HMMs$ genuinely accounts for stock market dynamics.

II. Problem formulation

The purpose of this project is to model two different stocks with *Hidden Markov Models*. We will use the models to forecast the adjusted closing price 100 days into the future. These models will be evaluated by three different evaluation methods which includes a trading analysis which uses the

forecasted price as a template for a trading algorithm. We want to see if there are any difference between the two models and if it is possible to use the models for stock trading.

A. Datasets

We chose two different stocks, *Tesla, Inc.* and *the Coca-Cola Company* to model in this project. These two stocks were chosen to evaluate our models' capacity of modeling stocks of varying volatility. *Tesla, Inc.* represents the high volatility stock and *the Coca-Cola Company* represents the low volatility stock.

We used a simplified notion of stock trading where you buy and sell stocks relative closing price and therefore chose the daily adjusted closing price of both stocks as the data used in this project.

For both stocks we train our model on data between 2010-06-29 and 2018-09-29. We forecasted 100 days into the future and used the historical price of these dates to evaluate the model.

III. Hidden Markov Model

A. Theory

A Hidden Markov Model (HMM) is a stochastic process consisting of a Markov chain, which has a finite number of states X_n [7] interlinked with a stochastic process Y_n which are assumed to be dependent on the hidden process X_n. The fundamental difference between a HMM and a Markov chain is the unobservable, hidden, states which represents the underlying mood of the stock which we can not observe. The observable process Y_n is the actual stock price we can observe. These hidden states describe distinctive market moods, where each mood corresponds to its own distinct trend. The stock price can therefore be interpreted as a sequence generated by the different hidden states and thus the logical relationship between hidden states and observations can be learnt from data.

The process can be identified by a compact notation $\lambda = (A, B, \pi)$, where A is the transition matrix, whose elements $a_{ij} = P(i_{i+1} = j | i_t = i)$ are depicting the probability of a transition from one state to another. B embody the emission matrix, yielding the observation symbol probability $b_i(o_t)$, which pinpoints the probability of the observation o_t when in state i,such as $b_i(o_t) = P(o_t | i_t = i)$. In conclusion, π is the initial state distribution, such as π =$P(i_1 = i)$.

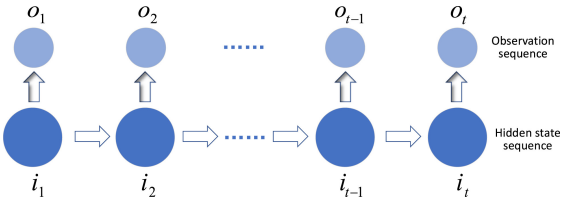


Fig. 1: Basic structure of Hidden Markov Model

Every interrelated hidden state sequence to an observation sequence $O = (o_1, ..., o_T)$ is expressed as $I = (i_1, ..., i_T)$, where $o_t = (o_t^1, ..., o_t^d)$, d is the dimension of observation value. Whereas the HMM is continuous, the emission probability is modeled as Gaussian mixture distributions,

$$b_i(o_t) = \sum_{k=1}^{K} c_{ik} g(o_t, \mu_{ik}, \Sigma_{ik}) \tag{1}$$

Where K is the number of Gaussian mixture components, c_{ik} is the mixture coefficient for the $k-th$ mixture in state i, $g(o_t, \mu_{ik}, \Sigma_{ik})$ is the multivariate Gaussian probability density function, where μ and Σ is the mean and covariance matrix for Gaussian distribution for state i [8]. Let $\alpha = \frac{1}{(\sqrt{2\pi}^d \sqrt{det(\sigma_{ik})})}$ and $\beta = exp[-\frac{1}{2}(o_t - \mu_{ik}\Sigma_{ik}^{-1}(o_t - \mu_{ik})^T]$, then

$$g(o_t, \mu_{ik}, \Sigma_{ik}) = \alpha\beta \tag{2}$$

Hence, a fully notation of the first-order HMM parameters, could be expressed as, $\lambda = \{\pi, A, c_{ik}, \Sigma_{ik}, i \in S\}$, where $S = \{0, ..., N-1\}$, note that N is the number of hidden states [9].

B. Training

For accurate and optimal use of an HMM, one have to consider the following questions:

- Considering a set of observations, what are the optimal model parameters?
- Considering a model and the corresponding set of observations, what is the optimal number of states?
- Considering a model, what is the likelihood to observe the considering set of data?

We use the $hmmlearn$ [10] package to solve the points above. The first point at issue is solved by using the Baum-Welch algorithm, which operates with Expectation-Maximization (EM) algorithm to reach for the optimal parameters for the HMM [11]. The third point at issue is solved by using the Forward algorithm [11]. For the second point at issue, we trained an arrangement of models by changing the number of states N. We varied N in a span from $[2, 15]$, where we calculated the negative log-likelihood of the training data which was in use for every of the models. The purpose of applying this was to choose the model which had the lowest value. Nonetheless, this method favors a more complex model, meaning that the number of states gravitated towards a higher number, which could result in overfitting. Afterwards, the performance of the identified sub-sequence is charted to the sub-sequence which is used for prediction. For the sake of eluding a more complex model, a penalty term to the negative log-likelihood was added. Based upon which of the penalty terms was chosen, restrictions were introduced on the model at a fluctuating degree. We analyzed two various performance measure metrics, especially, Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC). In BIC, the product of model parameters and the logarithm of the total of observation samples used was added, on the other hand, for AIC, the total of model parameters was added to the negative log-likelihood value, to attain the performance measure.

$$AIC = -2log(P(O_{trained}|\lambda)) - 2p \tag{3}$$

$$BIC = -2log(P(O_{trained}|\lambda)) - 2plog(T) \quad (4)$$

where $p = N^2 + 2N - 1$ and T is the number of observations. We have chosen the number of states, according to the performance measure of BIC.

C. Forecasting

Our implementation is based on the idea of calculating the log-likelihood of K preceding observations and correlating it with the log-likelihood of all the preceding sub-sequences of equivalent size by turning the time window by one day in the past. Note that K stands for latency and sub-sequences are sequences of equal latency as K. Next, we determine which day in the past whose log-likelihood of its K preceding observation is most correlated to the sub-sequence whose next day's price is to be predicted.

$$j = argmin_i(|P(O_t, ..., O_{t-K}|\lambda) - P(O_{t-i}, ..., O_{t-i-K}|\lambda)|) \quad (5)$$

where $i = 1, ..., \frac{d}{K}$. After that, we calculate the numerical price fluctuation from the selected day to its next day. This fluctuation is then added ongoing day's price to realize our next day's prediction.

$$O_{t+1} = O_t + (O_{t-j+1} - O_{t-j}) \quad (6)$$

Consequently, after we have attained the true observation, we incorporate it into the data set and recondition the model parameters in order to avoid model divergence. To put into perspective, the size of the sub-sequence is kept fixed, while another sub-sequence is located from the past data that displays an analogous pattern.

IV. Evaluation

Model evaluation is an essential part of the design process of a predictive model and due to the theme of this project we chose three different evaluation metrics to capture the performance of the model. We will be evaluating by statistical analysis, trend analysis with change point detection and trading analysis.

A. Statistical analysis

This part of the evaluation deals with statistical metrics as a tool to showcase the accuracy of the model. We have chosen three standard statistical metrics to evaluate the model which are seen below.

1) Mean Absolute Percentage Error (MAPE): The Mean Absolute Percentage Error (MAPE) is a statistical performance metric defined as the mean of absolute relative errors

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \quad (7)$$

where y_i is the actual value of index i, $\hat{y}_i$ is the forecasted value of index i and n is the length of the sequence.

2) Max Error (ME): The Max Error value is the maximum residual error of the sequence. A statistical metric showcasing the worst case error between the forecasted and true value.

$$ME = max(|y_i - \hat{y}_i|) \quad (8)$$

where y_i is the actual value of index i, $\hat{y}_i$ is the forecasted value of index i and $max()$ is function returning the maximum value of the sequence.

3) Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \quad (9)$$

where y_i is the actual value of index i, $\hat{y}_i$ is the forecasted value of index i and n is the length of the sequence.

B. Trend analysis

This part of the evaluation deals with how well the model manages to capture the seasonality of the actual data. This is done by change point detection through the `python package rupture` [12].

1) Graphical analysis: We used the search method `Dynp` [13] which uses a cost function to find the minimum of the sum of the costs of all subsequences of the time series. This is done over all possible segmentation.

The cost function used in the evaluation was `CostL1` and are based on the Least Absolute Deviation.

$$c(y_i) = \sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (10)$$

where $\{y_i\}_i$ is the signal on interval I $\hat{y}_i$ is the component wise median of signal $\{y_i\}_i$

We applied `Dynp` [13] on both datasets to be able to compare the graphical representation of the result. The red highlighted areas seen on the graphs are the detected change point areas and the blue areas are areas the algorithm have not detected any significant change in regards to the cost function the algorithm is following.

2) Numerical analysis: In the numerical part of the trend analysis we use two clustering metrics, Precision and Recall and the Rand Index.

The Rand Index (RI) is a measure of similarity between two data sets and determines the accuracy of the model. In our analysis the RI will be calculated on the change points segments detected by the method `Dynp` above. The Rand Index will produce a percentage as a measure of the similarity between the two change points segments and thereby the two data sets similarity in seasonality.

$$RI = \frac{TP + TN}{TP + FP + FN + TN} \quad (11)$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives and FN is the number of false negatives.

Precision and recall is a model evaluation metric using two measurements, Precision and recall, to determine the performance of a model.

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

where TP is the number of true positives and FP is the number of false positives.

Precision measures the correctly identified change points in relation to all identified change points, correct and incorrectly labeled. This will return a percentage of how precisely the modeled identified true change points.

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

where TP is the number of true positives, FP is the number of false positives and FN is the number of false negatives.

Recall measures the correctly identified change points in relation to change points correctly identified and change points not identified. This will return a percentage of how well the model managed to recall change points.

C. Trading Analysis

This part of the evaluation deals with the topic of trading. This will be done by applying a simple trading algorithm on our forecasted data. We will later compare the predicted gain against the actual gain or loss. This will show the reliability of the model when used as a basis for trading.

We will be using *Algorithm 1* which is a simple algorithm iterating through the list of forecasted prices, buying low and selling high. The algorithm will only trade when there is a perceived gain to the investment.

V. Results

TABLE I: Results of statistical analysis

	Tesla	Coca-Cola
MAPE	4.29	0.815
ME	7.580	2.24
RMSE	1.367	0.590

Notes: A lower value indicates better performance.

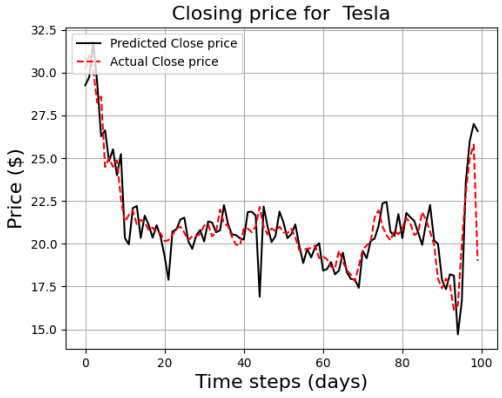
Algorithm 1 Algorithm for buying low and selling high

Input: list of forecasted prices
Output: calculates profit and buy/sell days
Initialisation :
$profit \leftarrow 0$
$minima \leftarrow 0$
$prices \leftarrow Input$
for $i = 0$ to $range(prices)$ **do**
  if $(prices[i-1] > prices[i])$ **then**
    $minima = i$
  end if
  if $prices[i-1] <= price[i]$ **and** $i + 1 \equiv len(price)$ **or** $price[i] > price[i+1]$ **then**
    $profit+ = (price[i] - price[j])$
  end if
end for
return $profit$

TABLE II: Results of numerical trend analysis

	Tesla	Coca-Cola
Recall	0.6	0.8
Precision	0.6	0.8
RI	0.95	0.94

Notes: All three metrics have a value between 0 and 1. 1 indicates the datasets are equal and 0 indicates totally difference. A higher value indicates better accuracy, precision or recall.



VI. Conclusion

A. The model

One would think that the result would be greatly affected by the choice of the model, for instance the number of states in Hidden Markov model, however that was not the case. When we implemented models with a higher number of hidden states the results didn't show an analogous change in the results. In conclusion, a more complex model with a higher number of states doesn't necessarily imply a better model.

B. Statistical analysis

The results from the statistical evaluation highlighted one of the major hypothesize we had in the beginning of the project

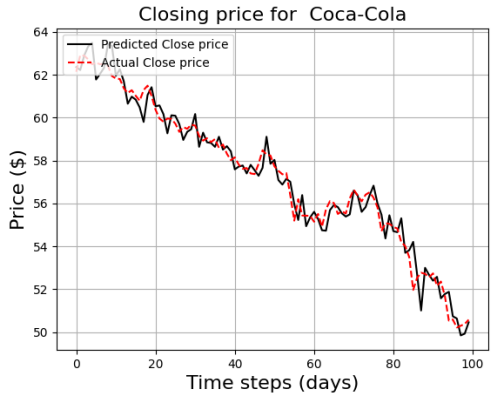


Fig. 2: Predicted closing price of Coca-Cola and Tesla

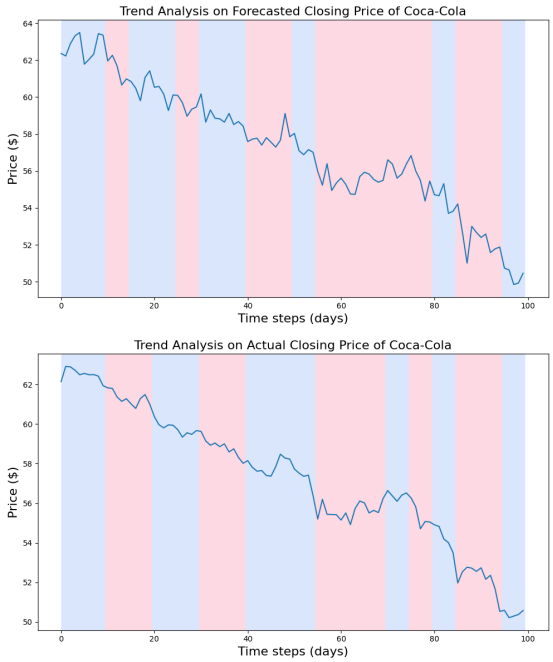


Fig. 3: The upper/lower graph shows how Algorithm 1 acted on the forecasted/actual price for Coca-Cola Company. The blue circle shows a buy and the green circle shows a sell. Every buy is followed by a sell and is considered a pair.

– would the volatility of the stock matter when modelling it? This question seems to be arbitrary, but due to the inherent risk of trading stocks, especially with volatile stock, it is crucial to evaluate the limitations of the model. As we can see from both *Figure 2* and *Table I* there are a tangible deviation in the *MAPE* and *RMSE* scores between the two models. The model based on the stock chosen as the high volatility choice, Tesla, Inc., have a higher metric on both counts and when considering the models are optimize regarding the models' parameters and uses the same time interval for the training dataset, it might indicate that the volatility of the stock does matter and could be used as a basis for future work.

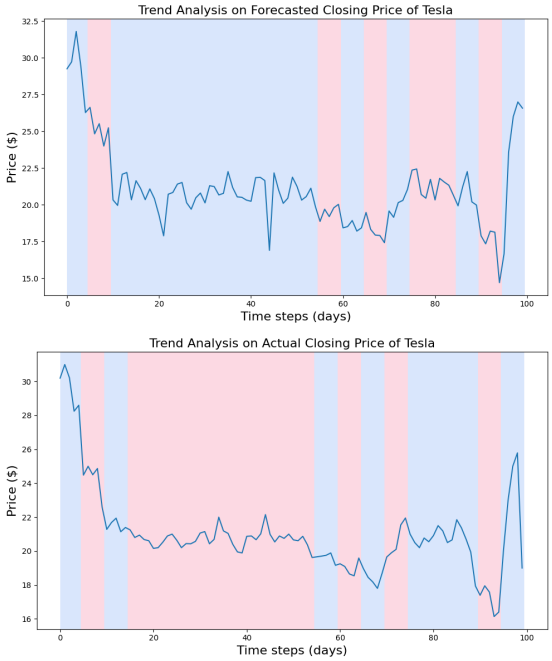


Fig. 4: The upper/lower graph shows how Algorithm 1 acted on the forecasted/actual price for Tesla, Inc. The blue circle shows a buy and the green circle shows a sell. Every buy is followed by a sell and is considered a pair.

C. Trend analysis

The results from the trend analysis seems to further strengthen the hypothesis of *the Coca-Cola Company model* being the better model in terms of statistical accuracy.

When discerning *Figure 3* we can see that the change point detection graphs of the *Coca-Cola Company* have detected areas of change points overlapping between the forecasted and actual price data. This should indicate that that the model is performing well and by looking at the numerical part of the trend analysis in *Table II* we see a RI score of *0,94* and Precision and Recall of *0,8*. These results indicates that the model managed to capture the inherent change point characteristics of the *Coca-Cola Company stock*. When discerning *Figure 4* there a major area of change points detected which are not capture by the model. This should indicate that the model in some ways have not been able to capture the inherent change point characteristics of the *Tesla, Inc.* stock. When looking at the numerical part of the trend analysis in *Table II* we see a RI score of *0,95* and Precision and Recall of *0,6*.

The Precision and Recall score of *0,6* looks reasonable considering the graphical representation of the trend analysis in *Figure 4*, but the RI score of *0,95* seems odd when you compare it to the score of the *Coca-Cola Company model*, which is smaller. This is what makes these two metrics needed together – even though the accuracy of the model is seemingly high, the Precision and Recall metrics shows that a more nuanced picture where the precision and recall are not as high as the *Coca-Cola Company model*. We have not found a good explanation for this instance and with this in mind we are

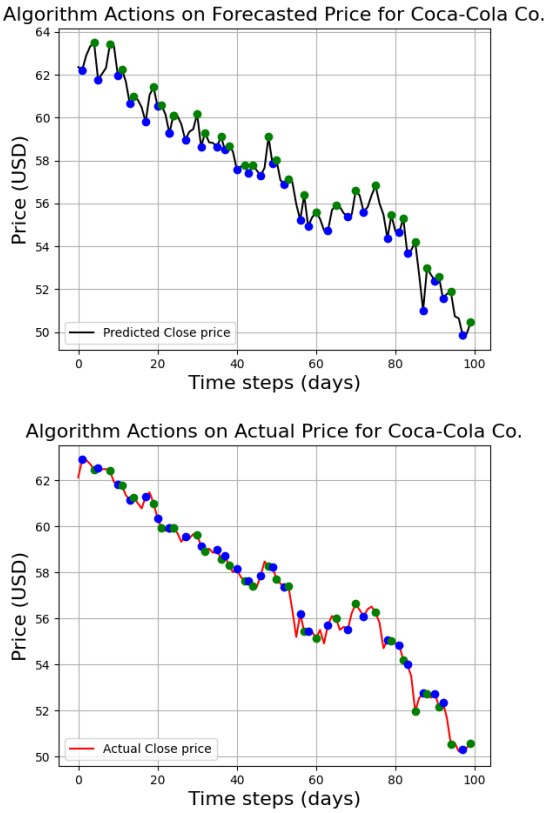


Fig. 5: The upper/lower graph shows how Algorithm 1 acted on the forecasted/actual price for Coca-Cola Co. The blue circle shows a buy and the green circle shows a sell. Every buy is followed by a sell and is considered a pair.

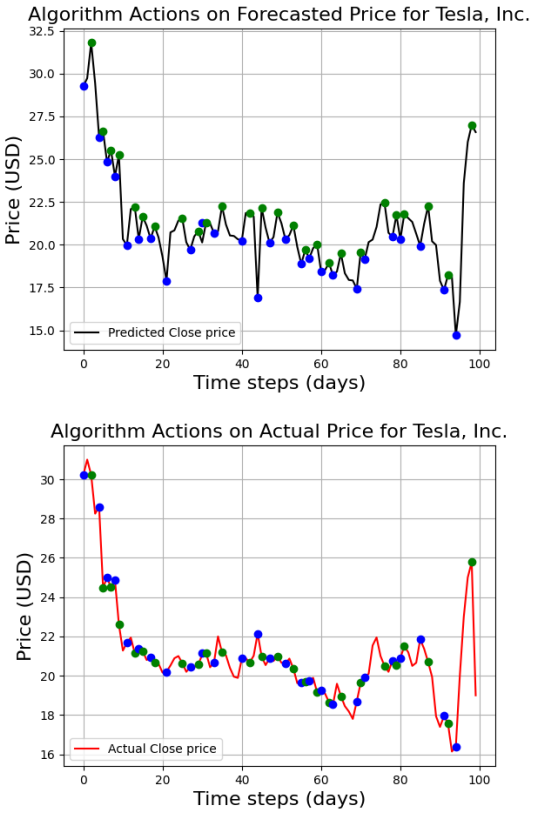


Fig. 6: The upper/lower graph shows how Algorithm 1 acted on the forecasted/actual price for Tesla, Inc. The blue circle shows a buy and the green circle shows a sell. Every buy is followed by a sell and is considered a pair.

not dismissing the RI score as a measure of accuracy, but it is needed to be accompanied with the Precision and Recall scores.

D. Trading analysis

In *Table III* we can see how *Algorithm 1* fares on the forecasted price of the *Tesla, Inc.* stock. It managed 11 trades with an actual gain out of the 25 trades with a perceived gain. That is a *44 percent success rate* when forecasting a trade with a perceived gain. The biggest discrepancy between a perceived gain and an actual gain is *26,4 percent*, meanwhile the biggest discrepancy between a perceived gain and actual loss is *36,5 percent*. The model did however manage to forecast Trade 25 in *Table III* which resulted in a trade with an actual gain of *57,3 percent*. Out of the 5 trades with the biggest perceived gain 4 trades were an actual gain, but out of these actual trade gains there were an average discrepancy of *16,4 (16,375) USD* between the perceived and actual gain.

In *Table VI* we can see how *Algorithm 1* fares on the forecasted price of the *Coca-Cola Company* stock. It managed 8 trades with an actual gain out of 28 trades with a perceived gain. That is a *29 percent success rate* when forecasting a trade. The biggest discrepancy between a perceived gain and an actual gain is *2,5 percent*, meanwhile the biggest discrepancy between a perceived gain and actual loss is *4,7 percent*.

The suitableness of *Algorithm 1* can be questioned and it might be the case that another algorithm would have fared better. That is a possibility and makes it difficult to draw any conclusion from this evaluation concerning the fitness of these models in stock forecasting. The simplicity of how *Algorithm 1* operates makes it a valuable indicator if the model is accurate. When buying low and selling high with no restrictions an accurate model is needed.

The discrepancy between the performance of the *Coca-Cola Company model* and the *Tesla, Inc.* model in the statistical and trend analysis and the trading analysis. One would have thought a model performing well in the prior two evaluation methods would also perform well in the third – this further highlight the inconclusiveness of the trading analysis.

E. Summary

As we can see from the results it is possible to model both stocks and as we hypothesized the stock of low volatility had better results on the statistical and trend analysis metrics. The results from the trading analysis is inconclusive and makes the hunt for a better performing *Hidden Markov Model* a future endeavour.

TABLE III: Results from trading analysis
Tesla

	Perceived gain	Actual gain/loss
Trade 1	+8,6	±0
Trade 2	+1,3	-14,4
Trade 3	+2,7	-2,0
Trade 4	+5,2	-9,3
Trade 5	+11,2	-12,5
Trade 6	+6,4	-0,7
Trade 7	+3,6	-1,2
Trade 8	+20,3	+2,2
Trade 9	+5,6	+0,6
Trade 10	+5,8	+0,4
Trade 11	+7,7	+2,4
Trade 12	+8,0	-1,0
Trade 13	+31,2	-5,3
Trade 14	+8,8	+0,5
Trade 15	+4,0	-1,3
Trade 16	+4,4	+0,2
Trade 17	+2,7	-3,0
Trade 18	+7,0	-3,2
Trade 19	+7,5	+2,2
Trade 20	+17,2	+5,1
Trade 21	+6,3	+3,0
Trade 22	+7,2	-1,1
Trade 23	+11,7	+2,8
Trade 24	+5,0	-5,3
Trade 25	+83,7	+57,3

Notes: The trades 1-25 consists of a buy and sell pair visible in Figure 5. Green numericals with a '+' in front means a gain, red numericals with a '-' in front means a loss and ±0 means approximately no gain nor loss.

TABLE IV: Results from trading analysis
Coca-Cola

	Perceived gain	Actual gain/loss
Trade 1	+2,0	-1,0
Trade 2	+2,7	-0,3
Trade 3	+0,5	-0,1
Trade 4	+0,5	+0,2
Trade 5	+2,7	-0,5
Trade 6	+0,06	-0,7
Trade 7	+1,4	±0
Trade 8	+2,1	+0,1
Trade 9	+1,1	-0,4
Trade 10	+0,8	-0,7
Trade 11	+0,3	-0,7
Trade 12	+0,3	-0,9
Trade 13	+0,7	-0,5
Trade 14	+3,2	+0,7
Trade 15	+0,3	-0,9
Trade 16	+0,4	+0,1
Trade 17	+2,1	-1,4
Trade 18	+1,2	-0,5
Trade 19	+2,2	+0,5
Trade 20	+2,2	+2.0
Trade 21	+2,2	+0,3
Trade 22	+2,0	±0
Trade 23	+1,2	-1,5
Trade 24	+0,9	-3,8
Trade 25	+3,9	±0
Trade 26	+0,3	-1,1
Trade 27	+0,6	-3,5
Trade 28	+1,2	+0,5

Notes: The trades 1-28 consists of a buy and sell pair visible in Figure 5. Green numericals with a '+' in front means a gain, red numericals with a '-' in front means a loss and ±0 means approximately no gain nor loss.

VII. Future Work

We would like to further investigate the hierarchical Hidden Markov Model which uses a hierarchical structure of multiple Hidden Markov Models. This model have been shown to successfully model stocks with a high accuracy when using different scales, coarse and fine, of time intervals in the different layers. It would be interesting to apply the hierarchical structure to the same trading analysis as in this project to see if it would perform better.

VIII. Acknowledgement

We want to thank our supervisor Yu Wang for putting up with all our questions, rescheduling and mishaps.

References

[1] T. Cazenave and S. B. Hamida, "Forecasting financial volatility using nested monte carlo expression discovery," in *2015 IEEE Symposium Series on Computational Intelligence*, 2015, pp. 726–733.

[2] Y. Lin, H. Guo, and J. Hu, "An svm-based approach for stock market trend prediction," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, 2013, pp. 1–7.

[3] D. M. Q. Nelson, A. C. M. Pereira, and R. A. de Oliveira, "Stock market's price movement prediction with lstm neural networks," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 1419–1426.

[4] M. R. Hassan and B. Nath, "Stock market forecasting using hidden markov model: a new approach," in *5th International Conference on Intelligent Systems Design and Applications (ISDA'05)*, 2005, pp. 192–196.

[5] S.-H. Park, J.-H. Lee, J.-W. Song, and T.-S. Park, "Forecasting change directions for financial time series using hidden markov model," in *Rough Sets and Knowledge Technology*, P. Wen, Y. Li, L. Polkowski, Y. Yao, S. Tsumoto, and G. Wang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 184–191.

[6] N. Nguyen, "Hidden markov model for stock trading," *International Journal of Financial Studies*, vol. 6, no. 2, 2018. [Online]. Available: https://www.mdpi.com/2227-7072/6/2/36

[7] Z. Lu, "Hidden markov models for time series: An introduction using r, 2nd edition, by walter zucchini, iain l. macdonald, and roland langrock. monographs on statistics and applied probability 150, published by crc press, 2016. total number of pages: 28+370. isbn: 978-1-4822-5383-2 (hardback)," *Journal of Time Series Analysis*, vol. 39, 09 2017.

[8] D. C. Scott L. Miller. (2012) Probability and random processes (second edition). [Online]. Available: https://www.sciencedirect.com/topics/mathematics/gaussian-probability-density-function/

[9] M. Zhang, X. Jiang, Z. Fang, Y. Zeng, and K. Xu, "High-order hidden markov model for trend prediction in financial time series," *Physica A: Statistical Mechanics and its Applications*, vol. 517, pp. 1–12, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437118314018

[10] D. Tsai, "Unsupervised learning and inference of hidden markov models," [Online]. Available from: https://hmmlearn.readthedocs.io/en/latest/tutorial.html, May 25 2021.

[11] N. Nguyen, "An analysis and implementation of the hidden markov model to technology stock prediction," *Risks*, vol. 5, no. 4, 2017. [Online]. Available: https://www.mdpi.com/2227-9091/5/4/62

[12] C. Truong, L. Oudre, and N. Vayatis, "Selective review of offline change point detection methods," *Signal Processing*, vol. 167, p. 107299, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165168419303494

[13] E. Paris-Saclay. (2017) Exact segmentation: dynamic programming. [Online]. Available: https://centre-borelli.github.io/ruptures-docs/user-guide/detection/dynp/

Asynchronous Advantage Actor-Critic and Flappy Bird

Markus Fredriksson and Marcus Wibrink

***Abstract*—Games provide ideal environments for assessing reinforcement learning algorithms because of their simple dynamics and their inexpensive testing, compared to real-world environments. Asynchronous Advantage Actor-Critic (A3C), developed by DeepMind, has shown significant improvements in performance over other state-of-the-art algorithms on Atari games. Additionally, the algorithm A3C(lambda) which is a generalization of A3C, has previously been shown to further improve upon A3C in these environments. In this work, we implement A3C and A3C(lambda) on the environment Cart-Pole and Flappy Bird and evaluate their performance via simulation. The simulations show that A3C effectively masters the Cart-Pole environment, as expected. In Flappy Bird sparse rewards are present, and the simulations reveal that despite this A3C manages to overcome this challenge the majority of times, achieving a linear increase in learning. Further simulations were made on Flappy Bird with the inclusion of an entropy term and with A3C(lambda), which display no signs of improvement in performance when compared to regular A3C.**

***Sammanfattning*—Spel utgör ideella miljöer för att bedöma reinforcement learning algoritmer på grund av deras enkla dynamik och billiga testning jämfört med verkliga miljöer. Asynchronous advantage actor-critic (A3C) utvecklad av DeepMind har visat betydande förbättringar på Atari spel jämfört med andra etablerade RL-algoritmer. Vidare har algoritmen A3C(lambda), som är en generalisering av A3C, tidigare visats ge ännu bättre resultat för dessa spel. I denna studie implementerar vi A3C och A3C(lambda) på miljöerna Cart-Pole och Flappy Bird och utvärderar algoritmerna via simulering. Simuleringarna visar att A3C på kort tid bemästrar Cart-Pole, som väntat. I Flappy Bird är användbar information glest fördelad och belöningen har ett lokalt optimum vilket leder till att algoritmen riskerar att fastna. Trots detta visar simuleringarna att A3C lyckas ta sig förbi det lokala optimat majoriteten av försöken och förbättrar sin belöning linjärt därefter. Ytterligare simuleringar gjordes på Flappy Bird genom att inkludera en entropiterm och med A3C(lambda). Ingen av metoderna visade någon märkbar förbättring jämfört med vanlig A3C.**

***Index Terms*—reinforcement learning, A3C, entropy, A3C(lambda), Cart-Pole, Flappy Bird, sparse rewards.**

Supervisors: Damianos Tranos

TRITA number: TRITA-EECS-EX-2021:146

I. Introduction

A common way for humans to learn is to interact with their environment, in a cause and effect manner, without having explicit guidance. For example, humans learn to walk using this approach. In Reinforcement Learning (RL), this style of learning from interaction is done computationally [1, p.1]. Games are one of the most commonly used environments for RL. This is due to the fact that they are artificially created and thus the dynamics are both known and modifiable. Furthermore, they provide complex environments with an infinite supply of useful data and can be simulated much faster than real-world interactions [2, p.1].

One such game environment is the mobile game Flappy Bird which gained a lot of popularity in early 2014 [3]. Flappy Bird is a 2D side-scroller where a bird is flown between obstacles, and the game is known for being notoriously hard and addictive. We found Flappy Bird to be an interesting environment for evaluating the RL method Asynchronous Advantage Actor-Critic (A3C) introduced by DeepMind [4]. In their article, DeepMind shows that A3C outperforms the previous state-of-the-art algorithms in Atari domains, while training for significantly less time.

The main purpose of this study is to implement A3C, and measure its performance with a handful of simulations. Another important aspect of this study is to gain personal knowledge in RL, and simultaneously present a self contained article on the subject to share this knowledge with the reader.

II. Theory

Reinforcement learning is a field concerned with creating an intelligent agent which is able to make optimal decisions in an environment with respect to maximizing some cumulative reward. The agent, an RL algorithm in charge of decision making, functions without prior knowledge of the environment and operates on its own through trial and error. Thus, no labeled test data is necessary and agents can be successful even when previous knowledge of the system is limited.

A. Markov Decision Processes

All RL problems consist of an agent and an environment. The agent is the decision maker which takes actions based on the current state of the environment. The environment represents the system, i.e. everything that affects the outcome of the agent's actions. Markov Decision Processes (MDPs) are a mathematical framework used to model RL problems and serve as a basis for RL theory.

An MDP is a discreet-time stochastic process in which state-transition probabilities are influenced by an action, and every transition yields a reward. The system consists of a state-space S, an action-space A, a bounded scalar reward determined by $R : S \times A \to \mathbb{R}_0^+$, and a transition probability $p(s'|s, a)$ which is the probability of transitioning to state s', given state s and action a. The dimensions of A and S can either be finite or infinite. An MDP is entirely characterized by this 4-tuple (S, A, R, p), and we refer to it alternatively as the system or model.

An important aspect of MDPs is the Markov property. The Markov property holds for a system if the future of the system is only dependent on the current state, and not on the previous ones, i.e. $P(s_t|s_{t-1}, a_{t-1}, ..., s_0, a_0) = P(s_t|s_{t-1}, a_{t-1})$. All environments are assumed to behave like MDPs which allows this mathematical framework to be applied on any RL problem.

B. Choosing Actions

In order to choose the right actions, the following is necessary: policy, reward, value function and model. The algorithm chooses actions according to a policy $\pi(a|s)$, which is a probability distribution of different actions the agent can take from that state. After every action it receives a reward, and the objective of the algorithm is to maximize its discounted return, that is the accumulated rewards with a discount factor included. The discounted return is defined as $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$, where γ is the discount factor. The purpose of the discount factor is to prioritize events in the near future [1, p.55]. The value function under the policy π is a mapping $v_\pi : S \to \mathbb{R}$, and it is defined by $v_\pi(s) = \mathbb{E}[G_t|S_t = s]$, for all $s \in S$. The rewards are collected from the states the agent visits using the policy it has in the current state s_t. More intuitively, the value function is the expected sum of the discounted return, when starting in a given state and following the policy thereafter. An optimal policy implies an optimal value function v_π^*, i.e. in any given state the value function is maximized. The model is the agent's estimation of the environment dynamics. Model-based algorithms directly estimate the MDP of the environment, whereas model-free algorithms learn the value function or the policy directly. Most well-known deep RL algorithms such as DQN [5], DDPG [6], and A3C [4], are model-free.

C. Policy Gradient Methods

In policy gradient methods, a function form of the policy π is parametrized by some parameter $\theta \in \mathbb{R}^d$, $d \in \mathbb{N}$. The function is tuned in order to increase the probability of actions that lead to a higher return, and decrease the probability of actions that lead to a lower return, until the optimal policy is reached. This can be achieved by changing θ in the direction of the gradient of some scalar performance measure $J(\theta)$, called the objective function. The objective function can have any form as long as it differentiable with respect to θ, and it measures the performance of the policy.

The Policy Gradient theorem states that for an objective function J, the gradient of J is equal to the policy gradient

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s, a) Q_\theta(s, a)], \tag{1}$$

where $Q_\theta(s, a)$ is the action-value function parametrized by θ, defined by $Q_\theta(s_i, a_i) = r_i + V_\theta(s_{i+1})$. If the policy is not used to sample actions, J does not depend on θ and the gradient becomes zero for all θ. Thus, all Policy Gradient methods have to use the policy to sample actions, i.e. they are all 'on-policy'.

D. Actor-Critic in Policy Gradient

In (1), the action value function $Q_\theta(s, a)$ needs to be estimated using the rewards from an entire episode. This causes high variance which can lead to slow convergence. Actor-critic methods reduce the variance by introducing a critic $Q_w(s, a) \approx Q_\theta(s, a)$, which is a parameterization of the action-value function with respect to $w \in \mathbb{R}^{d'}, d' \in \mathbb{N}$. The actor updates the policy parameter θ in the direction suggested by the critic. Equation (1) can now be approximated as

$$\nabla_\theta J(\theta) \approx \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s, a) Q_w(s, a)]. \tag{2}$$

The critic allows for using batches of steps that don't extend over the entire episode at the cost of introducing bias.

E. Advantage

To further reduce variance in the policy gradient methods, a baseline function $B(s)$ can be subtracted from the policy gradient in (2). In order to not introduce further bias, it is required that the policy gradient of the baseline is zero. Thus, the baseline can always be moved into the policy gradient. Subtracting the baseline from (2) and moving it inside of the policy gradient yields

$$\nabla_\theta J(\theta) \approx \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s, a)(Q_w(s, a) - B(s))]. \tag{3}$$

A good choice of a baseline function is the value function $V_v(s)$, parametrized by $v \in \mathbb{R}^{d''}, d'' \in \mathbb{N}$. With the value function as the baseline function, the policy gradient i.e. (3) takes the form

$$\nabla_\theta J(\theta) \approx \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s, a)(Q_w(s, a) - V_v(s))]. \tag{4}$$

The last term in (4) can now be written as

$$A(s, a) = Q_w(s) - V_v(s), \tag{5}$$

which is defined as the advantage function. To avoid the usage of two different function approximators for evaluating the advantage function, the TD-error $\delta_v = r + \gamma V_v(s') - V_v(s)$ is introduced, where s is the current state, and s' is the next state. The expected value of the TD-error is the advantage

$$\begin{aligned} \mathbb{E}_v[\delta_v|s, a] &= \mathbb{E}_v[r + \gamma V_v(s')|s, a] - V_v(s) \\ &= Q_v(s) - V_v(s) \\ &= A_v(s, a), \end{aligned}$$

and thus it is an unbiased estimate of the advantage function. Therefore, the advantage function can be parametrized with only v, and inserting (5) into (4) yields

$$\nabla_\theta J(\theta) \approx \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s, a) A_v(s)]. \tag{6}$$

F. Asynchronous

In asynchronous methods, the agent does not directly interact with the environment. Instead, it employs several workers who each have their own instance of the environment with which they interact. All workers explore the environment in parallel, providing the agent with asynchronous updates of the parameters. The purpose of this is to break the correlation that is introduced when training batches include data from

consecutive steps. A classical way of decorrelating the data is to use experience replay [5, p.5] which takes a random sample from a large batch of data. However, experience replay uses a large amount of memory and it requires that the algorithm is off-policy. Policy gradient methods are always on-policy and this precludes the usage of experience replay.

III. Asynchronous Advantage Actor-Critic

In this section the Asynchronous Advantage Actor-Critic algorithm (A3C) is introduced. The A3C combines all the topics discussed in subsections C-F to one single algorithm, hence the name.

A. Implementation

The pseudo code for the algorithm is presented as Algorithm 1. Our implementation of the algorithm can be viewed here: https://github.com/Wibrink/A3C-and-Flappy-Bird.

Algorithm 1 Asynchronous advantage actor-critic - pseudocode for each actor-learner thread. [4, p.13]

// Assume global shared parameter vectors θ *and* θ_v *and global shared counter* $T = 0$
// Assume thread-specific parameter vectors θ' *and* θ'_v
Initialize thread step counter $t \leftarrow 1$
repeat
  Reset gradients: $d\theta \leftarrow 0$ and $d\theta_v \leftarrow 0$.
  Synchronize thread-specific parameters $\theta' = \theta$ and $\theta'_v = \theta_v$
  $t_{\text{start}} = t$
  Get state s_t
  repeat
    Perform a_t according to policy $\pi(a_t|s_t;\theta')$
    Receive reward r_t and new state s_{t+1}
    $t \leftarrow t + 1$
    $T \leftarrow T + 1$
  until terminal s_t **or** $t - t_{\text{start}} == t_{\text{max}}$
  $R = \begin{cases} 0 & \text{for terminal } s_t \\ V(s_t, \theta'_v) & \text{for non-terminal } s_t \text{// Bootstrap from } s_t \end{cases}$
  for $i \in \{t-1, \ldots, t_{\text{start}}\}$ **do**
    $R \leftarrow r_i + \gamma R$
    Accumulate gradients wrt θ' :
      $d\theta \leftarrow d\theta + \nabla_{\theta'} \log \pi(a_i|s_i;\theta')(R - V(s_i;\theta'_v))$
    Accumulate gradients wrt θ'_v :
      $d\theta_v \leftarrow d\theta_v + \partial \left(R - V(s_i;\theta'_v)\right)^2 / \partial \theta'_v$
  end for
  Perform asynchronous update of θ using $d\theta$ and of θ_v using $d\theta_v$.
until $T > t_{\text{max}}$

The first **repeat** clause in the pseudo code is repeated for every new episode. An episode is a single interaction with the environment which lasts until a terminal state is reached. A terminal state is defined as an inescapable state with a reward of zero. t is a step counter which is incremented once per step within each episode. At the beginning of every episode, the gradients are reset and t is set to an initial value $t = t_{\text{start}}$. Furthermore, the local parameters are updated to have the same values as the global ones $\theta' = \theta$ and $\theta'_v = \theta_v$. In the second **repeat** clause, the episode is run through **until** a terminal state is reached or the step count t reaches t_{max}, a specified hyperparameter. The return of the final state needs to reflect all future rewards. Therefore it is set to $R = 0$ if the final state is terminal, as there is no future reward to be gained. Otherwise it is approximated with the value function estimate $R = V(s_t, \theta'_v)$ since the value function is defined as the expected discounted future return. In the **for** loop the n-step return is calculated for every step in the training batch. Finally, the actor- and critic-loss is calculated and the global parameters are updated by the gradients with respect to the local parameters. The function approximators chosen for the actor and the critic are neural networks.

B. Neural Networks for Actor and Critic

An illustration of the architecture for the neural networks is presented in Figure 1. The neural networks are implemented

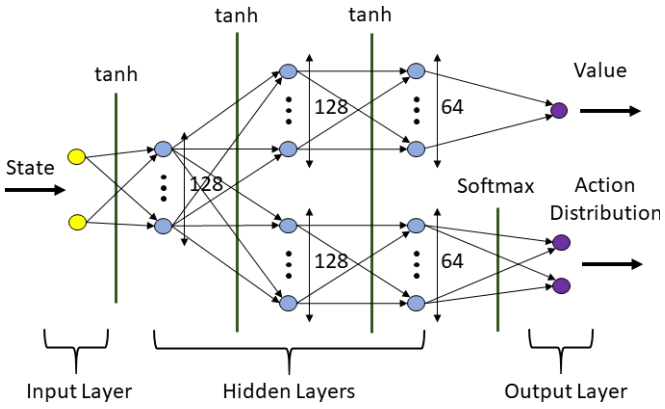


Fig. 1. The neural networks for the actor and critic. The yellow layer is an input layer, the blue hidden layers, and the purple output layers. The green lines are activation functions between layers and the two networks share the same input layer as well as the first hidden layer.

for the parameters θ and θ_v with two common layers. The input for the neural networks is the state and the output is an action distribution for the actor and an estimate of the value function for the critic. The neural networks have three hidden layers where the first one is shared, and the remaining two layers are equally distributed for the actor and critic. The sizes of the layers are presented in Figure 1.

C. Hyperparameters

The A3C algorithm has several hyperparameters that can be tuned in order to customize it for each environment.

The learning rate is a gradient scaling factor. A higher learning rate increases the values of the gradients, which speeds up learning but increases the variance. A lower value slows down learning but decreases oscillatory behaviour and makes the algorithm more stable.

The discount factor scales down rewards that are far into the future. A discount factor of zero causes the agent to ignore all future rewards, picking whatever action gives the most immediate reward. As the discount factor increases, the algorithm gains an increasing amount of foresight.

For every step a worker executes in an environment, it will save the experience from that step in a batch. The batch size t_{max} represents how many steps that need to be taken before the worker trains on the batch. The larger the batch size,

the more rewards are taken into account during training and the fewer states have to be estimated with the value function approximation. A larger batch size allows for training with more new experiences at the cost of correlating the training samples. The correlation is due to the steps being consecutive and therefore dependent on each other.

The number of workers represent how many workers are run in parallel. The more workers that are run, the more diverse the collected experiences become and many workers decrease the correlation of the data. If many workers are used in relation to the amount of computational cores, however, the algorithm will slow down. The ranges for the different parameters are shown in Table 1.

For each environment, different hyperparameters are examined. The learning rate and discount factor regulate the trade-off between stability and variance as well as greed and foresight. The batch size and amount of workers regulate the trade off between efficiency and decorrelation as well as efficiency and computational speed. For each experiment, hyperparameters which give a good trade-off between the different characteristics are chosen based on trial and error procedures.

TABLE I
HYPERPARAMETERS

Parameter	Range	Description
α	$\mathbb{R}^+$	Learning Rate
γ	$[0, 1]$	Discount factor
$t_{\max}$	$\mathbb{N}^+$	Batch size for training
w	$\mathbb{N}^+$	Number of workers

D. Measuring Performance

In order to measure the performance of the algorithm on an environment, some useful measurable quantities need to be introduced. Since the goal of the agent is to accumulate as much reward as possible during an episode, the total reward, that is the sum of rewards, is a good measure of the performance. The total reward is expected to increase as the agent improves and thus directly shows the performance of the algorithm.

Another measure of interest is whether the value function has converged. The easiest way of measuring this is to pick a set of points in the state-space. The points in the set are always the same and they are distributed across the space. Depending on how the value function parameterization changes, the changes will be reflected in different domains of the state-space. If these points are well distributed, any change in the value function parameterization will alter the value in at least one of the points. Then the average of the value function over these points is a good measure of the change in the value function parameterization. If this quantity is constant across episodes, the value function is most likely not changing and has converged.

Since both the environment and the policy are stochastic, every unique run of the algorithm, every realization, will yield different results. The agent might be very successful in one realization and then completely fail in the next one. Consequently, the performance has to be averaged over many realizations in order to gain any accuracy. Since all A3C workers play the game independently and thus generate one measure each per episode and per realization, the quantities are averaged for every worker across all realizations. Thereafter, since every worker reflects the performance of the agent equally, the quantities are averaged over every worker as well, generating a single graph of the mean quantity averaged over workers and realizations, as a function of episodes.

This yields two graphs of interest: the mean total reward across episodes, and the mean value function across episodes.

IV. CART-POLE

The main target of our project is the game Flappy Bird. However, in order to ascertain that our implementation of A3C is functional and in order to assess its performance, we will test the algorithm on a simpler environment, namely Cart-Pole. The Cart-Pole environment is chosen because episodes can be simulated fast and it is easy for the agent to master. Another reason is that it has no local maximum which could lead to sub-optimal convergence. For these reasons, the Cart-Pole environment is a standard choice for testing in RL.

A. Environment

In Cart-Pole, a pole is fastened on a cart with a joint making it an inverted pendulum, and the goal is to balance the pole. The pole is allowed to deviate up to 15 degrees from the vertical axis, after which the game will end. In addition, the cart needs to be within a specific region the whole time. In order to balance the pole, the cart is pushed either to the left or right. The amount of steps that the agent can balance the pole is regarded as the total reward, with a maximum of 200. Figure 2 illustrates the environment.

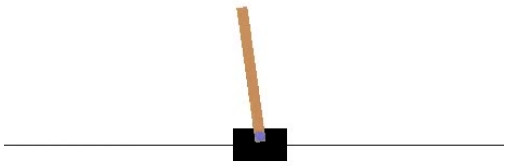


Fig. 2. The CartPole environment.

B. Simulation

The algorithm A3C, discussed in section III, is applied on the Cart-Pole environment. The following hyperparameters are used: $\alpha = 10^{-4}$, $t_{\max} = 10$, $\gamma = 0.99$ and $w = 8$. Furthermore, 20 realizations are simulated and every realization contains 1000 episodes per worker. The neural network architecture follows the one described in section III.

C. Results

The results are presented in two figures: the mean total reward and the mean value function, as described in section III subsection D. The mean total reward is the average total

reward across several realizations across all workers and is a good measure of the performance of the algorithm. The mean value function figure is the average value of the value function taken over a fixed set of points in the state-space, averaged over all realizations and workers. This is a measure of the convergence of the value function. The greyed out area represents the values within one standard deviation of the mean.

In Figure 3, the mean total reward increases drastically during the first 200 episodes, after which it starts to stagnate, increasing slowly. It seems to stop improving at a reward of about 185. Furthermore, the variance is greater during the initial growth phase and then decreases.

In Figure 4, the value function initially increases until about 200 episodes after which it initially drops and then starts to fluctuate. After about 600 episodes, it decreases drastically and then proceeds to fluctuate again. The stagnation at 200

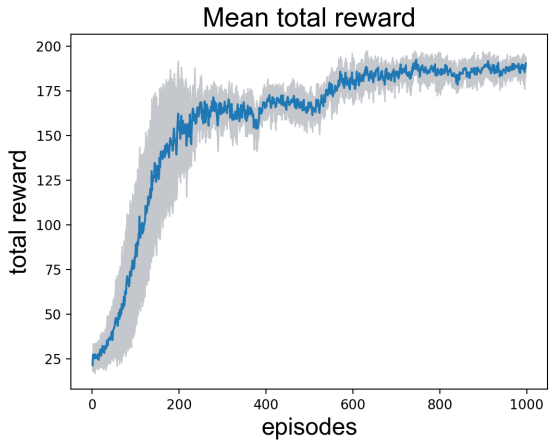


Fig. 3. The average total reward across 20 realizations across all workers. The majority of learning takes place during the first 300 episodes, after which it stabilizes on an average total reward of 185.

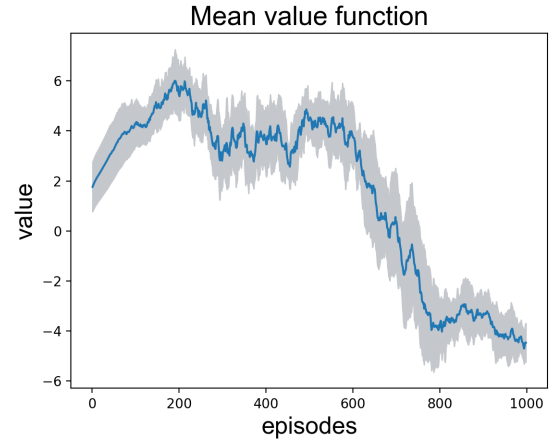


Fig. 4. The mean value function approximation averaged over a set of states across 20 realizations across all workers. The fluctuating curve shows no sign of convergence.

episodes is likely due to the fact that the agent starts to approach the maximum attainable reward. For the mean score to reach a certain value, it is acceptable that some instances are below the average as long as some are above it. When the mean reward approaches 200, however, no instances can be higher than 200. This means that the only way for the mean to reach a reward of 200 is if no instance is below 200, which is harder to achieve. It is likely that the mean total reward would continue to increase at the same speed as in the beginning if the simulation would continue past 200 points.

Even though it might look like the algorithm has converged based on the mean total reward, the fluctuations of the mean value function indicate otherwise. As stated, the agent improves slower as the reward approaches 200, but as of 1000 episodes, it has not yet stopped changing its value function and policy. This means that the algorithm has not converged yet and the agent is still improving, albeit slowly.

The mean value function starts decreasing after 200 episodes when the mean total reward starts to stagnate. Both of the events are likely related to the agent reaching a score of 200 for the first time. As the mean total reward increases, the mean value function increases since the agent can gain more future rewards due to an improving policy. The agent estimates that it can keep increasing its mean total reward indefinitely. When it eventually does reach a total reward of 200 it realizes that it can not improve indefinitely since there is a cap to the total reward. Therefore it realizes that it has overestimated the values of the states and consequently, the mean value function now decreases to compensate.

The decrease in variance of the mean total reward over time is likely due to the fact that even though different realizations may yield different results, the agents converge towards similar policies over time. Thus, the more episodes that have passed, the closer the realizations are to converging and consequently to each other, which causes the variance to decrease.

D. Conclusion

Overall the simulations on Cart-Pole show that the A3C algorithm works well, and it is convincing that our implementation of A3C is correct and functional. The stagnating behaviour is likely caused by the environment, rather than the algorithm, and the decrease in variance of the mean total reward indicates that the different realizations converge towards similar policies.

Since the algorithm has shown promising results on the Cart-Pole environment, it is natural to further investigate its performance on the Flappy Bird environment which is the main target of this project.

V. Flappy Bird

The mobile game Flappy Bird is a 2D sidescroller where the player controls a bird flying between obstacles which consist of pipes with gaps in between them. The gaps appear at random heights and touching any part of a pipe or the ground ends the game. The goal is to pass as many pipes as possible before touching anything and thus ending the game. The final score is the amount of pipes passed through.

The bird moves forwards on its own at a constant speed. The screen can be tapped in order for the bird to flap its wings

and gain a fix amount of height, repeated taps can thus be used to gain more height. If no action is taken the bird descends downwards. The final score is the amount of pipes passed. An illustration of the Flappy Bird is presented in Figure 5.

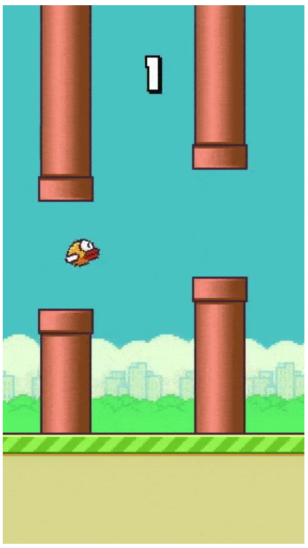


Fig. 5. The bird is passing through the first pipe in the mobile game, Flappy Bird.

A. Environment

The environment used for testing is FlappyBird-v0 from OpenAi Gym [7]. In this implementation of Flappy Bird, the state-space is continuous and consists of two spatial dimensions: the distance to the next pipe, and the difference between the bird's height and the height of the next gap. The action space is two-dimensional and discreet consisting of two actions: flapping and doing nothing.

On every step, the environment applies a constant downwards acceleration on the bird until a terminal velocity is reached. Thus, if the bird does nothing, it will eventually hit the floor and the game will end. Flapping instantaneously replaces the bird's horizontal velocity with a fixed upwards velocity, independent of its previous value. Since acceleration is applied on every step, flapping once will cause the bird to travel in an arc eventually falling down again.

The world has a lower boundary, the ground, but lacks an upper one. Thus the bird can fly up outside the screen. Since the gaps in the pipes are always on-screen, however, this would result in the bird hitting the next pipe unless it manages to come back down in time. The gap size is fixed and the gap heights are generated randomly. Furthermore, the distance between consecutive pipes is constant but the distance from the start position to the first pipe is longer.

In this environment, the agent receives a constant reward of 1 for every step the bird stays alive rather than for every pipe it passes. Passing through the first pipe is equivalent to accumulating a total reward of 111. Each pipe passed after that point adds an average of 37 additional reward to the total reward.

B. Simulation

The algorithm A3C is applied on the Flappy Bird environment. The following hyperparameters are used: $\alpha = 2 \cdot 10^{-4}$, $\gamma = 0.99$, $t_{\text{max}} = 30$ and $w = 8$. The neural network architecture stays unmodified.

C. Results

The results are presented in two figures: mean total reward and mean value function, same as in the Cart-Pole simulation. In Figure 6, the variance increases as the episodes increase, and the total reward seems to stay constant, after which it increases linearly. The value function increases drastically to 40, and slowly decreases thereafter. The constant line

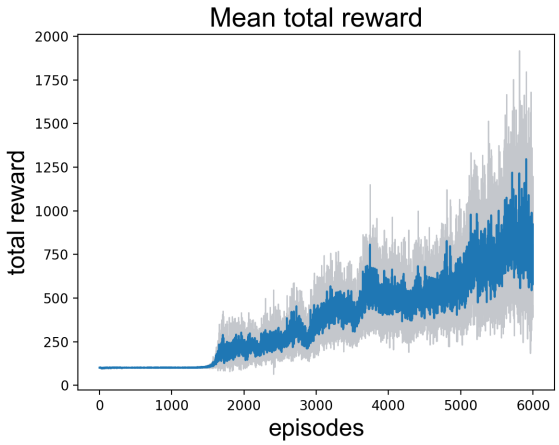


Fig. 6. The mean total reward is constant over the first 1700 episodes, after which the agent manages to fly through the first gap and the learning begins to increase linearly. A total reward of 111 corresponds to a score of 1, and for every additional reward of 37, an additional score of 1 is received.

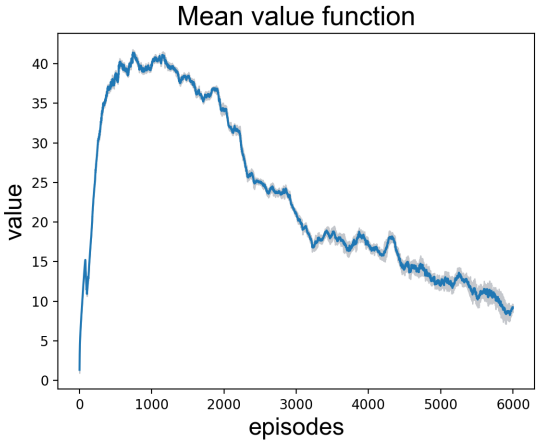


Fig. 7. The mean value function increases drastically in the beginning, where the agent struggles to get through the first gap. After the first pipe is passed, the estimate of the value function decreases as the states are found to be less valuable than previously estimated.

in Figure 6 is the phase where the agent struggles to get through the first gap, which is one of the difficulties in the environment. On average the agent manages to get through

the first gap after approximately 1700 episodes, but over the 20 realizations, the agent passes the first pipe only 14 times, which translates to a success rate of 70%. The high variance in the figure is a consequence of this. It can be the case that given enough episodes, the agent always manages to get through the first gap. This would imply that it is only an issue of slow convergence. However, it can also be the case that it converges to a local maximum. This means that the agent believes that the best strategy is to avoid crashing to the ground, which leads to constantly overshooting the first gap. This issue can also be observed in Figure 7. The drastic increase in the beginning is an over estimate of the value function, which indicates that the value function over estimates the value of the states where the agent overshoots the first gap. When the first pipe is passed, the value function drops since the agent notices that crashing into the first pipe is undesired.

After it manages to get through the first gap, the agent begins to learn the environment, which can be seen from the linear increase in Figure 6. The fact that the total reward is not converging, stems from the fact that there is no cap on the total reward, making it an unsolved environment. However, even if the environment is unsolved, an optimal policy can be found, which leads to convergence in the value function. There is no indication of convergence in the value function, which is expected as 6000 episodes is not sufficient to reach an optimal policy. Therefore, it is not of much interest to see how high total rewards the agent manages to obtain. It is of more interest that the total reward increases linearly, which indicates that the agent is learning the environment. It can still be noted that after 6000 episodes it manages to obtain a total reward of approximately 900, on average, which corresponds to a score of 22 in the game. This is comparable to human performance, and given enough episodes, it can be speculated that super human performance can be reached, due to the linear increase in total reward.

D. Conclusion

From the simulations on the Flappy Bird environment it is clear that 6000 episodes is not enough to master the environment. However, this is enough to reveal a possible short coming in the A3C algorithm. The fact that the agent only managed to get through the first gap 70% of realizations means that either A3C has slow convergence in this environment or it gets stuck in a local maximum. Attempts at making the algorithm more reliable in the environment are addressed in the following sections.

VI. Entropy

In this section an attempt to fix the issue in Flappy Bird is discussed, with an implementation of the entropy term as suggested in [4, p.4]. The entropy term encourages exploration, which could lead to an escape of the possible local maximum where the agent overshoots the first gap with the only intention to avoid hitting the ground. The entropy term acts as noise and makes the policy more stochastic, which in turn leads to variation in the actions which the agent takes. The entropy term is added to the objective function (6), which yields the modified objective function

$$\nabla_\theta J(\theta) \approx \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a) A_v(s) + \beta H(\pi_\theta(s))],$$

where $H(\pi_\theta(s))$ is the entropy of the policy distribution for a given state and β is a hyperparameter, the entropy regularization factor.

With the entropy term added to A3C, the same hyperparameters are used. The results are the following: It does not solve the issue, because in order to pass through the first gap, the agent can only flap around 10% of the times. The entropy term only makes it harder to maintain the flapping frequency low enough, due to the effect of making the policy more stochastic. In other words, the entropy encourages the agent to overshoot the first gap even more, which is not desired.

VII. A3C(λ)

In order to address the issue of slow convergence caused by sparse rewards in the Flappy Bird environment, a generalized version of A3C called A3C(λ) [8] is implemented. The idea of A3C(λ) is to improve the sample efficiency [8, p.1], which is a measure of how effectively the agent utilizes past experiences during training. This can lead to a significant increase in convergence speed, which is the main issue with the A3C algorithm in the Flappy Bird environment. In A3C(λ), a quantity called λ-returns, R^λ, are calculated in addition to the n-step returns. Recall that in A3C, as seen in the pseudo code, the n-step returns are calculated by the recursive formula

$$R_{i+1} = r_i + \gamma R_i.$$

Furthermore, in A3C there is a trade off between 1-step learning, and n-step learning. The 1-step learning uses the reward in the current state, and estimates the states that follow with the current value function. The problem with 1-step learning is the fact that it only uses one reward in the training batch, which leads to slow convergence. In n-step learning n rewards are used in training which improves the learning speed because more information is available. However, with larger values of n, the variance may be higher and more samples are required compared to 1-step learning. The λ-returns serve as a balance between 1-step learning and n-step learning, combining the best aspects of both extremes. The λ-returns are defined as the exponential average of all n-step returns from the current step and onward [8, p.2], and are calculated recursively [8, eq.4]

$$R_i^\lambda = r_i + \gamma\big[\lambda R_{i+1}^\lambda + (1-\lambda) V(\hat{s}_{i+1})\big],$$

where $\lambda \in [0,1]$. Furthermore, the estimate of the state is exact, i.e $\hat{s} = s$ when the environment can be modeled as an fully observable MDP, which is the case in Cart-Pole and Flappy Bird. Note that if $\lambda = 1$ the λ-return is equal to the n-step return, which reduces A3C(λ) to A3C. Thus, A3C is simply a special case of A3C(λ).

Algorithm 2 A3C(λ) [8, p.6]

// Assume global shared parameter vectors θ *and* θ_v *and global shared counter* $T = 0$
// Recall that $\forall \hat{s} \in \mathcal{S},\ (\hat{s}) \implies V(\hat{s}; \theta_v) = 0$
repeat
  Reset local counter $t \leftarrow 0$
  Reset state $\hat{s}_0 \leftarrow \phi(o_0)$
  repeat
    Execute action $a_t \sim \pi(\cdot|\hat{s}_t; \theta)$
    Receive reward r_t and new observation o_{t+1}
    Approximate state $\hat{s}_{t+1} \leftarrow \phi(o_0, ..., o_{t+1})$
    $t \leftarrow t + 1$
    $T \leftarrow T + 1$
  until $(\hat{s}_t)$ **or** $t == t_{\max}$
  $R^\lambda = R = V(\hat{s}_t; \theta_v)$
  for $i \in \{t-1, ..., 0\}$ **do**
    $R \leftarrow r_i + \gamma R$
    $R^\lambda \leftarrow r_i + \gamma\big[\lambda R^\lambda + (1-\lambda)V(\hat{s}_{i+1}; \theta_v)\big]$
    Improve actor:
      $\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi(a_i|\hat{s}_i; \theta)\big[R - V(\hat{s}_i; \theta_v)\big]$
    Improve critic:
      $\theta_v \leftarrow \theta_v - \alpha \nabla_{\theta_v}\big[R^\lambda - V(\hat{s}_i; \theta_v)\big]^2$
  end for
until $T \geq t_{\max}$

A. Implementation

The pseudo code for A3C(λ) is presented as Algorithm 2. The λ-returns are calculated in the same **for** loop as the n-step returns. The actor is updated using the returns R as in A3C, but the critic is updated using the λ-returns. Apart from this, the algorithm is identical to A3C.

B. Advantages over A3C

Using lambda-returns allows for actor-critic algorithms, such as A3C, to reap the benefits of eligibility traces, even though these two methods are normally incompatible [8, p.1]. Furthermore, variance-reducing properties of the λ-returns allow for larger values of $t_{\max}$ [8, p.5]. This enables the algorithm to benefit more from the lower bias of Monte-Carlo based methods, while maintaining a lower variance. Overall, adding lambda-returns to A3C has proven to be effective when learning to play Atari 2600 games [8, p.1]. Therefore it is probable that adding a lambda-return to the A3C algorithm can enhance learning in the Flappy Bird environment.

VIII. A3C(λ) on Flappy Bird

In this section A3C(λ) is applied on the Flappy Bird environment in order to study whether this algorithm handles the starting phase better, where the A3C agent has trouble getting through the first gap. The same hyperparameters are used as with A3C: $\alpha = 2 \cdot 10^{-4}$, $\gamma = 0.99$, $t_{\max} = 30$, $w = 8$. Additionally, using trial and error procedures, a value of $\lambda = 0.4$ was picked after showing promising results.

The number of realizations are 20, and 6000 episodes per worker is used for every realization. The architecture for the neural network is not altered in this experiment.

A. Results

The results for the Flappy Bird environment are presented below. The set of figures are the same as in the Cart-Pole simulation. Similarly to the normal A3C algorithm, A3C(λ)

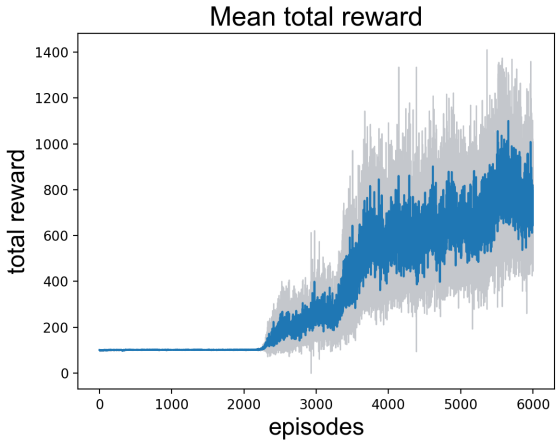


Fig. 8. The mean total reward across 20 realizations averaged over all workers. It is constant over the first 2300 episodes after which it rapidly increases. A reward of 111 corresponds to a score of 1, and for every additional reward of 37, an additional score of 1 is received.

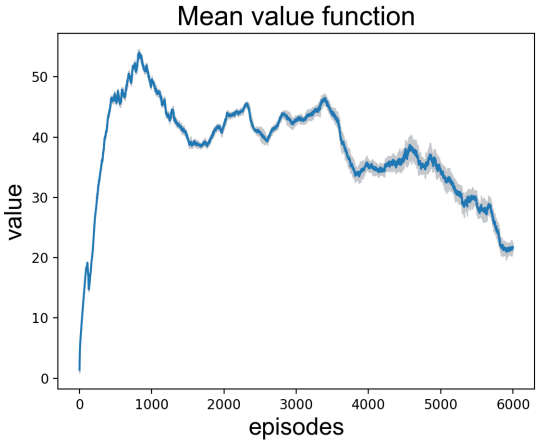


Fig. 9. The mean value function across 20 realizations averaged over all workers. The critic initially overestimates the value of some states within a local optimum. After it passes the first pipe it realizes that the values of the states were overestimated and reduces its estimations.

initially struggles to get through the first gap. As seen in Figure 8, after about 2300 episodes, the agents start to get a higher reward indicating that they have gone past the first pipe. After that point, the mean total reward steadily increases.

Figure 9 shows no sign of convergence. This is expected since the environment has no upper reward limit and can be played indefinitely. The rather high variance observed in the mean total reward figure can be explained by the fact that in certain realizations, the first pipe cannot be surpassed. This clusters the data into two groups: the ones who are stuck at the bottom, and the rest. Additionally, only 10 out of 20 realizations managed to pass the first pipe, which translates to a success rate of 50%. It can however be observed that the

agents do increase their total reward once they manage to find the way through the first gap.

In contrast to the normal A3C algorithm, the value function experiences a drop before the total reward starts to increase. This is likely caused by some agents partially surpassing the pipe, but without properly learning how to consistently repeat it. This can occur before the observed increase in total reward at 2300 episodes and might lead to the value function starting to account for the gap, but without enough experience to consistently get through it. This also explains why no major drop of the value function can be observed at 2300 episodes. Additionally, it implies that A3C(λ) has a more difficult time learning, even when it partially passes the first pipe. Together with the fact that fewer realizations managed to get through the first pipe, one can conclude that A3C(λ) is less stable than A3C when simulating Flappy Bird.

The total reward at 6000 episodes is only about 800 compared to the 900 for normal A3C. This corresponds to a score of 19 compared to 22 for A3C. However if the fact that the total reward started increasing at a later point is considered, the growth after that point does not differ noticeably.

B. Conclusion

The results from A3C(λ) are similar to A3C. The success rate did however decrease from 70% to 50%. This indicates a loss in consistency. Furthermore, the algorithm requires more episodes to get through the first gap. Both algorithms increase their total rewards about the same amount per episode once they get past the first pipe however. Overall, A3C(λ) seems less reliable.

IX. Conclusion

The simulation on Cart-Pole shows that A3C works well and is able to reach an average total reward of 185 within 800 episodes. Moreover, the convergence of the algorithm is further strengthened by the decrease of variance over time.

The simulation on Flappy Bird shows that A3C is capable of learning more complex environments, reaching a score of 22 within 6000 episodes, albeit with some issues when encountering sparse rewards. A3C manages to overcome this for the majority of times, with a success rate of around 70%. Several attempts for improving the success rate are implemented. Including the entropy term does not increase the success rate, as a matter of fact, it decreases. This is due to the fact that entropy encourages the policy to remain stochastic, which further encourages suboptimal behaviour in the Flappy Bird environment. Enhancing A3C to the A3C(λ) algorithm shows a similar outcome with λ set to 0.4, reaching a score of 19 within 6000 episodes, and a success rate of 50%. A decrease in success rate is unexpected, as A3C(λ) has been shown in the past to have superior performance, compared to A3C.

A. Accuracy of Results

Since both the environments and the algorithms are stochastic, a successful result does not necessarily equate to a functioning method. Therefore it is always important to conduct several simulations in order to show that the results were not simply a happy coincidence. All of our simulations are averages over 20 independent simulations and even though A3C seems functional, more simulations would be necessary to be completely certain. If we had more time, we would have conducted more simulations and longer ones, with more episodes. This could show that the realizations stuck at the first pipe also converge given enough time. We can conclude that A3C does work, and likely, fairly reliably as well. But more extensive testing is very welcome.

A3C(λ), although worse than A3C, show similar results. The algorithm does seem somewhat stable and shows similar growth rate to A3C once it gets past the sparse rewards problem. In this study, only one value of λ was extensively tested. Although $\lambda = 0.4$ was picked for its promising performance during tuning, simulations of similar magnitude would have to be conducted to rule out that A3C(λ) is not better than A3C, for the right value of λ.

B. Future Studies

The main challenge in the Flappy Bird environment is to navigate the bird through the first gap. Even though A3C manages to do this the majority of times, more of a guarantee of this is desired. Due to limited time, only a handful of simulations were conducted, with limited amount of episodes. In future studies, new methods for addressing this issue can be tested. More specifically, methods that improve the convergence and methods that combats convergence to local maximum are of interest. Additionally, more extensive simulations of A3C on the Flappy Bird environment might show the algorithm to be more reliable than what is apparent here.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, England: The MIT Press, 2020.

[2] K. Shao, Z. Tang, Y. Zhu, N. Li, and D. Zhao, "A survey of deep reinforcement learning in video games," *CoRR*, vol. abs/1912.10944, Dec. 2019. [Online]. Available: http://arxiv.org/abs/1912.10944

[3] Artyom Dogtiev. (2020, June) Flappy bird revenue – how much did flappy bird make? Businessofapps, London, United Kingdom. [Online]. Available: https://www.businessofapps.com/data/flappy-bird-revenue/

[4] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," *CoRR*, vol. abs/1602.01783, Jun. 2016. [Online]. Available: http://arxiv.org/abs/1602.01783

[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing atari with deep reinforcement learning," *CoRR*, vol. abs/1312.5602, Dec. 2013. [Online]. Available: http://arxiv.org/abs/1312.5602

[6] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *CoRR*, Jul. 2019. [Online]. Available: https://arxiv.org/abs/1509.02971

[7] G. Nogueira. (2021, Feb) flappy-bird-gym. OpenAi, San Francisco, CA 94110. [Online]. Available: https://pypi.org/project/flappy-bird-gym/

[8] B. Daley and C. Amato, "Efficient eligibility traces for deep reinforcement learning," *CoRR*, vol. abs/1810.09967, Oct. 2018. [Online]. Available: https://arxiv.org/abs/1810.09967v1

Control of an Inverted Pendulum Using Reinforcement Learning Methods

Joel Kärn

***Abstract*—In this paper the two reinforcement learning algorithms Q-learning and deep Q-learning (DQN) are used to balance an inverted pendulum. In order to compare the two, both algorithms are optimized to some extent, by evaluating different values for some parameters of the algorithms. Since the difference between Q-learning and DQN is a deep neural network (DNN), some benefits of a DNN are then discussed.**

The conclusion is that this particular problem is simple enough for the Q-learning algorithm to work well and is preferable, even though the DQN algorithm solves the problem in fewer episodes. This is due to the stability of the Q-learning algorithm and because more time is required to find a suitable DNN and evaluate appropriate parameters for the DQN algorithm, than to find the proper parameters for the Q-learning algorithm.

***Sammanfattning*—I denna rapport används två algoritmer inom förstärkningsinlärning, nämligen Q-inlärning och djup Q-inlärning (DQN), för att balancera en omvänd pendel. För att jämföra dem så optimeras algoritmerna i viss utsträckning genom att testa olika värden för vissa av deras parametrar. Eftersom att skillnaden mellan Q-inlärning och DQN är ett djupt neuralt nätverk (DNN) så diskuterades fördelen med ett DNN.**

Slutstatsen är att för ett så pass enkelt problem så fungerar Q-inlärningsalgoritmen bra och är att föredra, trots att DQN-algoritmen löser problemet på färre episoder. Detta är pågrund av Q-inlärningsalgoritmens stabilitet och att mer tid krävs för att hitta ett passande DNN och hitta lämpliga parametrar för DQN-algoritmen än vad det krävs för att hitta bra parametrar för Q-inlärningsalgoritmen.

***Index Terms*—Reinforcement Learning, Q-learning, DQN, CartPole, Inverted Pendulum, OpenAI.**

Supervisor: Alexander Berndt

TRITA number: TRITA-EECS-EX-2021:147

I. Introduction

Machine learning is used in a wide range of areas to solve different types of problems. For some problems, an optimal solution is not necessarily known and it can be hard to define what needs to be done in order to obtain a wanted outcome. Reinforcement learning (RL) is a paradigm of machine learning that can be used for these types of problems [1]. An example of such a problem is the video game StarCraft II. StarCraft II is a real time strategy game where two players compete against each other. The goal is to destroy the other player's base while protecting your own. This is done by gathering resources, constructing buildings and forming an army. If a computer program was made to play Starcraft II, it would be hard to formulate a policy of what actions the program must take to win, since there is at all times an extremely large amount of actions that could be taken. In this case RL can be used to find a policy by letting a program explore the environment and from experience improving its policy each game. This has been done by DeepMind [2]. They created the program AlphaStar which was able to reach the highest competitive league of Starcraft II.

In this report, OpenAI's "CartPole-v0" [3] is investigated. Compared to Starcraft II, this game is much less complicated since it only consists of an inverted pendulum. A pole is attached to a cart that slides on a frictionless track. The inverted pendulum is controlled by either pushing the cart to the right or to the left. The goal of the game is to keep the pole from falling over. In this project, the RL algorithms Q-learning and deep Q-learning are used to play the game, where policies are formed that estimate the best actions to take in order to keep the pole upright as long as possible. The two algorithms are improved upon by evaluating some parameters and then the results are compared, to see which one performs the best in the environment and solves the game in the least amount of episodes, or playthroughs of the game.

II. Background

In this Section the relevant theoretical concepts and mathematics for reinforcement learning is included, and the CartPole environment is described further.

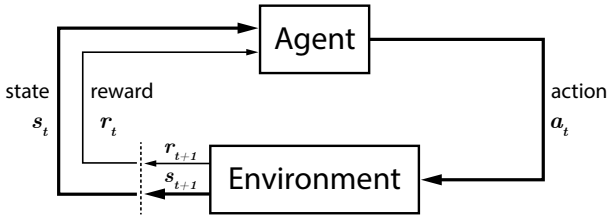


Fig. 1. Illustration of the reinforcement learning framework.

A. The Reinforcement Learning Framework

The reinforcement learning framework, seen in Figure 1, illustrates how RL works. An agent is in an environment at a certain state s_t. The state consists of the circumstances of the environment that the agent can observe at the given time step t. The agent then takes an action a_t which transitions the environment into a new state s_{t+1} and some reward r_{t+1} is given. The objective is to form a policy that the agent follows to get as much rewards as possible. Each action the agent takes will have some reward. The reward can be positive, negative or zero, and the best policy will over time accumulate the

highest amount of rewards. Finding a good policy is done by exploring the environment and exploiting the retrieved information. Exploration is done by pairing an action a_t with the state the agent is in s_t, and then recording the reward (r_{t+1}) that the action-state pair (a_t, s_t) results in. By storing many of these action-state pairs with their associated reward in a memory, they can then be used to form a policy. If the policy exploits rather than explores, it will select the action that is estimated to result in the most amount of future rewards.

B. The CartPole Environment

The game, or environment, that is used for this project OpenAI's "CartPole-v0" [3], which is a simulation of an inverted pendulum. A cart is attached to a frictionless track and a pole is fastened to to the cart, making an inverted pendulum. It is controlled by sliding the cart either to the right or to the left and in order to win the game the pole needs to be balanced for 200 time steps. If the pendulum tips over or the cart moves too far away from the center, the game is lost. One playthrough of the game is called an episode. By initiating the environment, the starting state is given as an output. With an action as the input, the environment outputs a next state and reward.

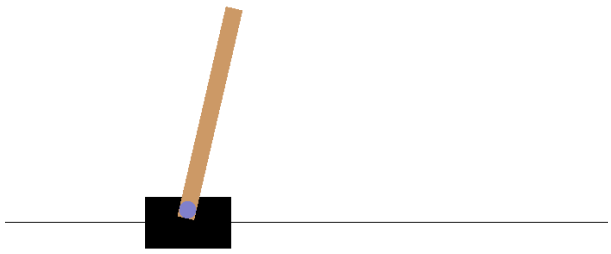


Fig. 2. Screenshot of one rendition of the CartPole-v0 environment.

Each state that is returned by the environment contains four values, namely the position of cart, the velocity of cart, the angle of the pole and the rotation rate of pole. This means that the state space of the environment is four-dimensional and that these four values are what the agent can observe of the environment. There are at each time step two different actions that can be taken. One is to push the cart to the right, and the other is to push the cart to the left. The action space is therefore two-dimensional. Each time step without the pole falling over or the cart moving too far from the center results in one extra reward. The maximum amount of rewards from one episode is therefore 200, since the game is won by reaching 200 time steps.

For this project, the inverted pendulum problem is considered solved if an algorithm runs at least 10 times and returns an average reward of 175 or more over 5 consecutive episodes. This is set up to prevent any one lucky run of an algorithm to solve the problem in very few episodes from happening. The less amount of episodes it takes for an algorithm to solve the problem, the better the algorithm is considered to be.

There is also the possibility to render the process as the game is played. This can be helpful for the viewer to understand the problem at hand but the algorithm's observation of a state is, like mentioned earlier, only four values. A screenshot of a rendition is shown in Figure 2, where a state is illustrated and the cart is dangerously far to the left while the pole is about to fall over to the right.

C. Mathematical Models

1) Q-Function: An agent's goal is to take the actions that cultivate the most total rewards as possible. By following a policy π, the future return G can be described as follows,

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + ... = \sum_{i=0}^{T} \gamma^i r_{t+i+1}, \quad (1)$$

where r_{t+i} is the reward of a time step $t+i$, γ is the discount factor and T is the last time step of the environment, which is 200 for the CartPole environment. The discount factor γ is in the range [0:1] and is used to decide how much future reward is valued. A low γ indicates the agent will value immediate rewards more than rewards in the distant future. Equation (1) can be described recursively in the following manner,

$$G_t = r_{t+1} + \gamma G_{t+1}. \quad (2)$$

Let S and A be defined as the state space and action space respectively. It would be convenient to have a function $Q^* : S \times A \rightarrow \mathbb{R}$ that takes any state-action pair s_t, a_t and returns a real value, called a Q-value, that measures the exact return from taking precisely action a_t in the state s_t. Q^* would then be the optimal Q-function and all actions in a certain state could then be valued and compared with their particular Q-values. In order for an agent to get as much rewards as possible, it could simply pick the action with the highest Q-value in each state. Such an optimal policy π^* is described as

$$\pi^* = \underset{a_t}{\text{maxarg}} Q^*(s_t, a_t). \quad (3)$$

However, not everything is known about the environment and Q^* and π^* are therefore unknown. Even so, a Q-function can be initiated with an arbitrary Q-value for each action-state pair and then be trained to resemble Q^*.

For this paper it suffices to mention that all Q-functions follow the Bellman equation [4],

$$Q^\pi(s_t, a_t) = r_t + \gamma Q^\pi(s_{t+1}, \pi(s_{t+1})), \quad (4)$$

where Q^π is the Q-function following policy π.

Each time an action is taken and the next state and reward are observed, the Q-function can be updated in the following recursive manner:

$$Q(s_t, a_t) = (1-\alpha)Q(s_t, a_t) + \alpha(r_t + \gamma \underset{a_{t+1}}{\text{max}} Q(s_{t+1}, a_{t+1})) \quad (5)$$

where α is the learning rate and γ is the previously mentioned discount factor. The learning rate decides how much the Q-function gets updated.

2) Bins for Q-learning: The states for the environment are not discrete. This means that the four values in s all vary between different limits and are continuous. Take for example the cart position. It can go be anywhere in the range $[-4.8, 4.8]$, which makes it impossible to set up state-pairs for all possible states that the inverted pendulum can be in. By discretization however, this is possible. Continuing with the example, the range is divided into four separate bins, $[-4.8, -2.4)$, $[-2.4, 0)$, $[0, 2.4)$ and $[2.4, 4.8]$. Now any state can be put into a bin. A relatively high number of bins will define the states more accurately but it will require more training for the Q-function to learn and for all Q-values to converge to their respective optimal values. The number of bins decides how many bins each state parameter has, meaning for example that if the number of bins is 10, all four parameters are divided into 10 bins each.

3) Epsilon-Greedy Method: The Epsilon-Greedy method randomly decides whether the agent explores or exploits. The value of epsilon (ϵ) is in the range $[0, 1]$. A random value between 0 and 1 is generated, and if this value is higher than ϵ, the agent takes the action that has the highest Q-value (exploitation) and if the random value is lower than ϵ, the agent takes a random action (exploration). If an agent only explores, it always takes random actions and it is unlikely to ever reach the highest amount of total rewards. If the agent only exploits it relies completely on previous experience and Q-values for some state-actions pairs might remain unexplored.

The value of ϵ should be high if much exploration is desirable and ϵ should be low if the agent is sufficiently trained and much reward is desired. For this project a linear ϵ decay function is used, which means that ϵ starts at 1 for the first episode and then decreases with a constant value until ϵ reaches 0 at a set episode. This is called the "ϵ limit" later in the report. Other variations of the Epsilon-Greedy method can be found in [5]

4) Deep Q-Network: A deep Q-network (DQN) is a deep neural network (DNN) [6] that is used to replicate a Q-function instead of a Q-table. DNN's consist of different amounts of layers that connect. A layer contains a set of neurons, or nodes, that connect to nodes in other layers via weights [6].

For the sake of this project, it suffices to note that a DNN can model a complex functions and if it is properly trained it can estimate Q-values for each state-action pair of an environment. Equations (3) and (4) are used for DQN's as well but instead of changing the Q-values of a Q-table, the weights of the DQN are updated instead, with backpropagation [7]. Much like the gradient of a function, the DQN is updated with gradient descent. A DQN algorithm is much like a Q-learning algorithm but since it simply estimates the Q-value from any action-state pair, the states do not need to be discrete.

The DQN for this project has four nodes in its input layer and two nodes in its output layer, in correspondence with the state space and the action space, with hidden layers between the input and output layers.

5) Replay Memory: Each time step t, an experience is saved as a tuple $e_t = (s_t, a_t, r_{t+1}, s_{t+1})$ in the replay memory D. A batch of random experienced from the replay memory is then used to update the DQN. This is done to in some extent stop the correlation between consecutive experiences. If for example the agent made some good decisions but a few bad ones made the pole fall over, the memory could be biased and a policy could be formed that estimates the good actions to be bad. By picking random actions out of the replay memory, such a bias can be avoided to some extent [8].

D. Algorithms

The two algorithms that were used in this project are Q-learning, presented in Algorithm 1, and DQN, presented in Algorithm 2. $T = 200$ is the maximum amount of time steps and M is the limit for the amount of episode. α and γ are the learning rate and the discount factor respectively and ϵ is from the Epsilon-Greedy method. D is the replay memory and N is its capacity, or the limit of how many experiences that can be stored in D.

Algorithm 1: The Q-learning algorithm

```
initialize α, γ and ϵ;
initialize Q-table with number of bins;
for episode = 1 to M do
    reset environment;
    for t = 1 to T do
        observe state s;
        generate random value ψ between 0 and 1;
        if ψ < ϵ then
            set a to random action;
        end
        else
            a ← max_{a_{t+1}} Q(s_t, a_t)
        end
        take action a;
        observe s_{t+1}, r_{t+1}, done;
        update Q-table;
        s_{t+1} ← s_t;
        if done then
            terminate episode;
        end
    end
end
```

III. METHOD

Since the goal of the project was to optimize both the Q-learning algorithm and the DQN algorithm to some extent, different parameters for the two algorithms were evaluated in order to find suitable values that would make the algorithms solve the problem as fast as possible.

A. Q-learning

To find an appropriate Q-learning algorithm, three parameters were evaluated, namely the learning rate (α), the discount factor (γ) and the number of bins. During the evaluation of these parameters, each simulation ran for 1000 episodes, and at least 10 simulations were run for each variation of a parameter.

Algorithm 2: The DQN algorithm

```
initialize α, γ and ϵ;
initialize replay memory D with capacity N;
initialize DQN;
for episode = 1 to M do
    reset environment;
    for t = 1 to T do
        observe state s;
        generate random value ψ between 0 and 1;
        if ψ < ϵ then
            set a to random action;
        end
        else
            a ← max_{a_{t+1}} Q(s_t, a_t)
        end
        take action a;
        observe s_{t+1}, r_{t+1}, done;
        store experience (s_t, a_t, r_{t+1}, s_{t+1}) in D;
        pick random experiences from D for batch
        update weights of DQN using batch;
        s_{t+1} ← s_t;
        if done then
            terminate episode;
        end
    end
end
```

The confidence interval of the lighter regions in the graphs was 100 percent, meaning that the upper limit of a lighter region shows the best result that was acquired and the lower limit shows an algorithm's worst performance. The epsilon decay was linear for the first 70 episodes. During the rest of the episodes, epsilon was zero.

Only three values for each parameter are included in each graph to make them more straightforward, but more values were tested. In each graph the best value and two worse values are included.

1) Choosing the Learning Rate: The learning rate is depicted as α in Equation (5). This parameter decides how much the Q-function gets updated each time step t. This means that a higher learning rate results in a faster change of the Q-function, but too high of a learning rate makes the process less stable and could result in overshooting and not converging to an optimal Q-function.

The three different learning rates that were investigated are shown in Figure 3.

As seen in Figure 3, an appropriate learning rate to be selected was 0.1.

2) Choosing the Discount Factor: The discount factor is written as γ Equation (5). The discount factor decides how much future rewards are valued compared to immediate rewards. A low discount factor makes the agent prefer sooner rewards over later rewards. If the discount factor is close to 1, then the agent seeks the rewards earned in the far future almost as much as the rewards in an immediate future.

Following the method of the last subsection, the same

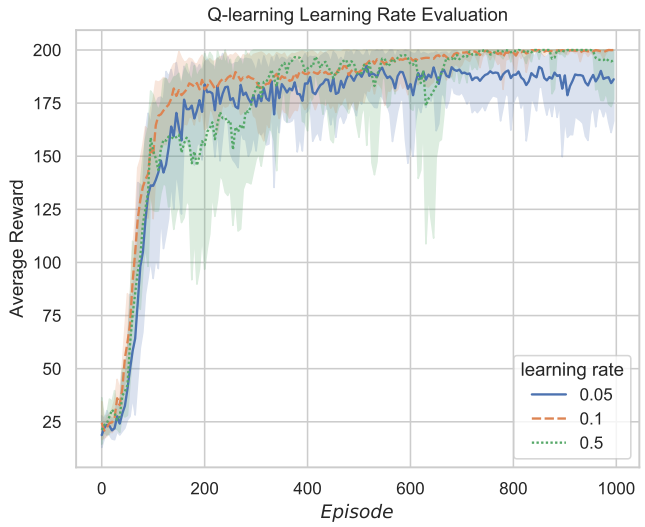


Fig. 3. Q-learning algorithm with different learning rates.

algorithm was tested with three different discount factors. The result is portrayed in Figure 4. In Figure 4 it is shown that a

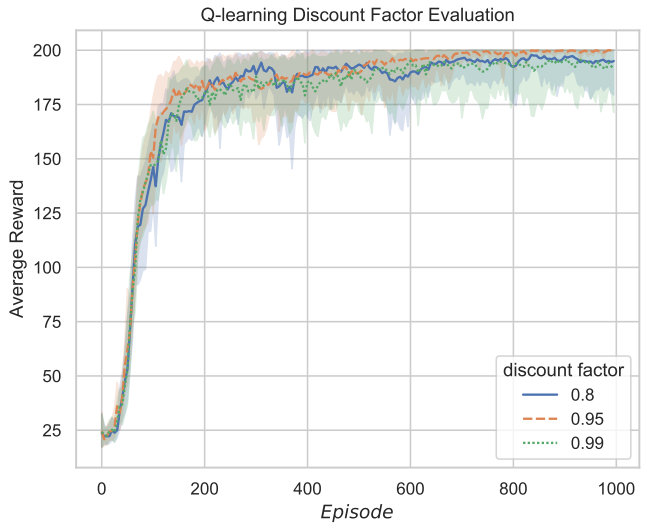


Fig. 4. Q-learning algorithm with different discount factors.

discount factor of 0.95 was appropriate.

3) Choosing the Number of Bins: The number of bins decides how detailed the Q-function is able to distinguish different states. A large number of bins will make the Q-function less discrete but will on the other hand require more experiences to train properly. Following the pattern of the previous subsections, the number of bins was evaluated by testing three different number of bins for the same algorithm. The result is seen in Figure 5.

From Figure 5, it is seen that 10 bins was a suitable amount for the algorithm.

4) Final Parameters for Q-learning: The final Q-learning algorithm was achieved with the values in Table I, put into Algorithm 1.

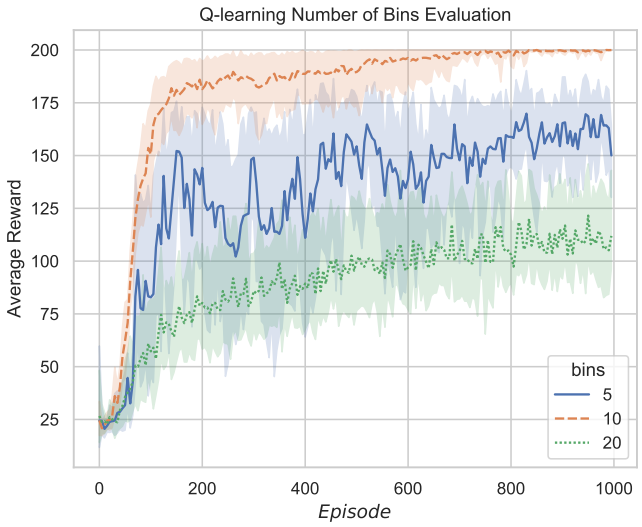


Fig. 5. Q-learning algorithm with different amounts of bins.

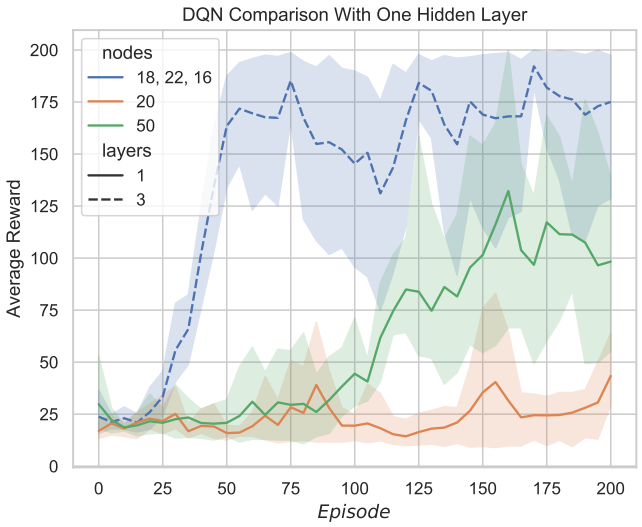


Fig. 6. DQN algorithms with a single hidden layer compared with three hidden layers.

TABLE I
Q-LEARNING PARAMETERS

Parameter	Value
learning rate (α)	0.1
discount rate (γ)	0.95
number of bins	10
ϵ limit ($episode$)	70
total episodes M	1000
runs	> 10

B. DQN

The DQN algorithm required much more computational time to gather the necessary data to plot informative graphs. It is therefore harder to draw conclusions from the data that is presented in the DQN part of this Section and the size of the DNN is the only evaluation presented in the report. Some differences between the Q-learning algorithm and the DQN algorithm are worth mentioning before presenting the graphs.

One big difference is, as recently pointed out, that the DQN algorithm took longer real time to run on the available hardware. This is because the DNN needed to be updated for each time step, instead of just updating the values of a Q-table. Therefore the amount of episodes for the plots was 200 instead of 1000. The discount factor remained the same as the one used for the optimized Q-learning algorithm, which was 0.95. This was because the relevance of previous action-state pairs should not depend on whether a DNN or a Q-table is used. No bins were evaluated since DQN did not require discrete action-state pairs. The learning rate was however changed to 0.001 because 0.1 resulted to be too unstable for the DQN algorithm. The learning rate was not evaluated further in this paper, instead only the amount of layers and the amount of nodes in each layer are presented in this subsection.

1) One Hidden Layer: First off one hidden layer was added between the input layer and the output layer. Different amount of nodes were tested, as seen in Figure 6.

As seen in Figure 6, no amount of nodes in a single hidden layer even managed to solve the problem in 200 episodes.

2) Two Hidden Layers: One more hidden layer was then added and the algorithm improved. Some different amount of nodes were selected and since the input layer had four nodes (this represents the state space of the environment) and the output layer had two nodes (this represents the action space of the environment), the first hidden layer was selected to be bigger than the second hidden layer. The result is portrayed in Figure 7.

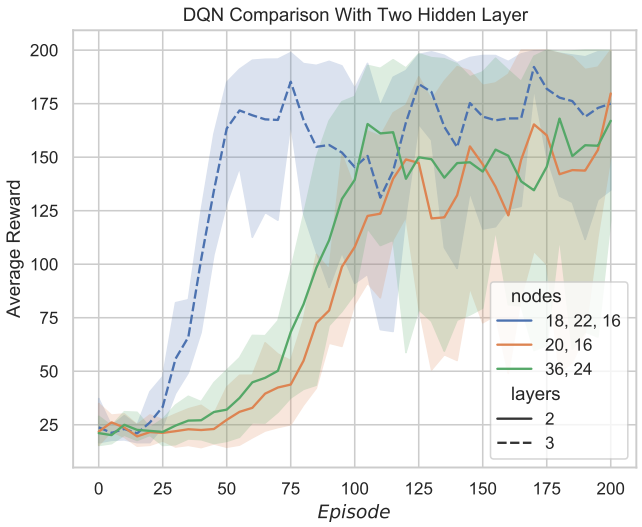


Fig. 7. DQN algorithms with two hidden layers compared with three hidden layers.

As seen in Figure 7, one added hidden layers improved the DQN algorithm, but the DQN with three layers was still the only one that managed to solve the problem.

3) Three Hidden Layers: One more hidden layer was added and this time the first hidden layer was larger than the last one,

like in the last Subsection, and the middle hidden layer was the largest. Three different variants of nodes have been selected and the result is displayed in Figure 8.

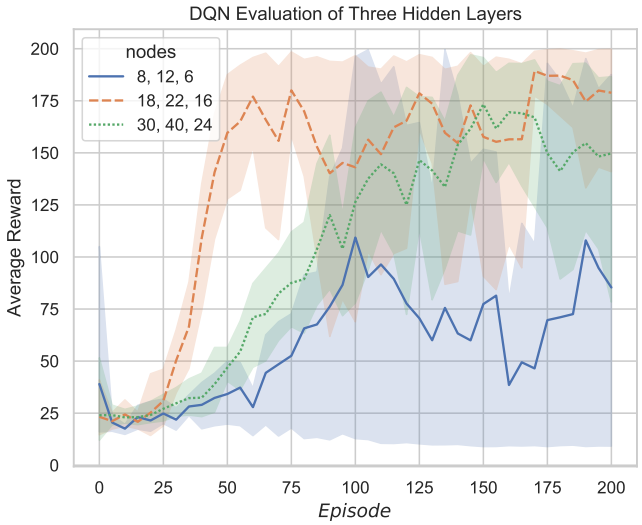


Fig. 8. DQN algorithms with different amount of nodes in three hidden layers.

Figure 8 shows that neither the bigger DQN or the smaller one managed to solve the problem. The final DQN was therefore selected to be the one with three hidden layers, with 18, 22 and 16 nodes.

4) Final Parameters for DQN: The final DQN algorithm was achieved with the values in Table II, put into Algorithm 2.

TABLE II
DQN PARAMETERS

Parameter	Value
learning rate (α)	0.001
discount rate (γ)	0.95
ϵ limit ($episode$)	70
hidden layers	3
nodes per layer	$18, 22, 16$
mini batch size	64
memory capacity N	2000
total episodes M	1000
runs	> 10

IV. RESULT

When the final DQN algorithm was found, it was run 11 more times for 500 episodes. It was then compared with the first 500 episodes of the final Q-learning algorithm. The result is shown in Figure 9.

In Figure 9, it is shown that the DQN algorithm managed to solve the problem in less than 80 episodes, while the Q-learning algorithm solved the task in more than 120 episodes, meaning that the DQN algorithm solved the problem in less episodes than the Q-learning algorithm. The confidence interval, or the light region of the graph, for Q-learning was however much smaller, meaning that the results from the algorithm were much less widespread and more reliable and stable.

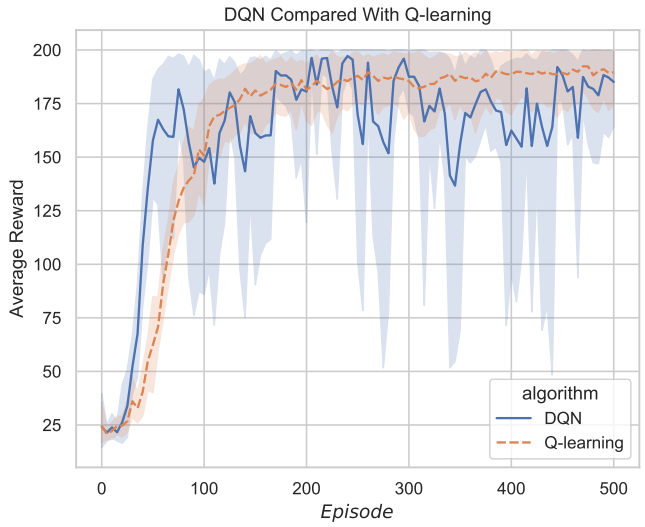


Fig. 9. Comparison between the Q-learning and DQN algorithms.

V. CONCLUSION

Throughout this project, a Q-learning algorithm and a DQN algorithm were optimized to some extent by evaluation of some of their parameters and they were then compared to one another.

It was found that the DQN algorithm solved the problem faster than the Q-learning, as seen in Figure 9. The DQN solved the problem in less than 80 episodes while it took the Q-learning algorithm more than 120 episodes to solve it. However, the result of the Q-learning algorithm was more stable. Referring back to Figure 9, it is seen that the DQN algorithm's lighter region band is much wider at places, which means the algorithm did not converge as well and took frequent dips. This indicates instability.

The obvious benefit of the DQN algorithm was that it solved the problem in less episodes than the Q-learning algorithm. Even though the solution was not as stable, the main requirement set up for this project was to solve the problem in the fewest amount of episodes, so in that regard the DQN was the better algorithm. Another benefit of the DQN algorithm was that the states did not need to be discretized. If the environment was not as well defined as this one, the Q-learning algorithm could be useless since it needed to be set up with a finite amount of action-state pairs. This makes a DQN very powerful compared to a Q-table, since it needs less information about the environment to work. Either the values of the action space or the state space can be discrete or continuous.

The benefit with the Q-learning algorithm was that it was easier to set up and run. Not much set up was required for the code to run without issues since the only thing being updated was the Q-table, which in code is just a matrix. The code also ran faster, which was an advantage when many different parameters were to be evaluated. As mentioned earlier, the Q-learning algorithm was also much more stable. The instability of the DQN was probably due to the fact that more evaluation

of the parameters like the learning rate (α) in the DQN was required to achieve similar stability. This was not investigated further in the paper.

ACKNOWLEDGMENT

The author would like to thank his supervisor Alexander Berndt for his guidance and incredible support throughout the project.

REFERENCES

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA: The MIT Press Cambridge, Nov. 2018, ch. 1, pp. 1–5.

[2] (2019, Oct.) AlphaStar: Grandmaster level in StarCraft II using Multi-Agent Reinforcement Learning. DeepMind, United Kingdom, London. [Online]. Available: https://shorturl.at/mDRZ7

[3] (2020, Mar.) Cartpole-v0. OpenAI, San Francisco, CA. [Online]. Available: https://gym.openai.com/envs/CartPole-v0/

[4] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA: The MIT Press Cambridge, Nov. 2018, ch. 3, pp. 62–68.

[5] ——, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA: The MIT Press Cambridge, Nov. 2018, ch. 2, pp. 28–31.

[6] K. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.

[7] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[8] R. Liu and J. Zou, "The effects of memory replay in reinforcement learning," in *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2018, pp. 480–482.

Warehouse Optimization by Multi-Agent Rollout Algorithms

Laura Briffa and William Emanuelsson

***Abstract*—Systems consisting of multiple robots are traditionally difficult to optimize. This project considers such a system in a simulated warehouse setting, where the robots are to deliver boxes while avoiding collisions. Adding such collision constraints complicates the problem. For dynamical multi-agent systems as these, reinforcement learning algorithms are often appropriate. We explore and implement a reinforcement learning algorithm, called multi-agent rollout, that allows for re-planning during operation. The algorithm is paired with a base policy of following the shortest path. Simulation results with up to 10 robots indicates that the algorithm is promising for large-scale multi-robot systems. We have also discussed the possibility of using neural networks and partitioning to further increase performance.**

***Sammanfattning*—System med flera robotar har traditionellt sett ansetts mycket svåra att optimera. I detta projekt undersöks ett sådant system i en simulerad lagerlokal, där robotarna skall förflytta lådor samtidigt som de undviker kollisioner. För dessa dynamiska system med flera robotar är förstärkande inlärning ofta lämpligt. Vi undersöker och implementerar en förstärkandeinlärningsalgoritm kallad ”multi-agent rollout” vilken möjliggör omdirigering under drift. Algoritmen används tillsammans med en så kallad ”base policy” som alltid väljer kortaste vägen. Baserat på simulationsresultaten med upp till tio robotar verkar algoritmen lovande för storskaliga flerrobotsystem. Det diskuteras även om möjligheten av att använda neurala nätverk och partitionering för att vidare öka prestandan.**

***Index Terms*—multi-agent problems, reinforcement learning, optimization and optimal control, collision avoidance, warehouse.**

Supervisor: *Yuchao Li*

TRITA number: TRITA-EECS-EX-2021:148

I. Introduction

Multi-agent systems consist of a set of agents, autonomous entities and decision makers, that are able to observe parts of their environment, coexist, and interact with each other [1]. Examples of multi-agent systems are robotic teams, traffic control systems, resource management systems, and multiplayer games [2], [3]. In dynamical environments, where the environment characteristics change with time, it is often inappropriate to predetermine agent behavior. Instead, the agents need to learn and adapt their behavior on-line [2].

Reinforcement learning (RL) algorithms can, among others, provide solutions for multi-agent problems. RL algorithms allows for an agent to learn without being told what action to take; it must itself discover what actions are most rewarding by trial and error [4]. Modern RL came to fruition in the late 1980's, when theories of trial and error learning originating from animal psychology combined with the theories of optimal control problems and dynamic programming. Several of the methods for solving optimal control problems, such as dynamic programming, are fundamental in modern RL algorithms and theory. Reasonably, RL problems are closely related to optimal control problems, and one could say that solution methods for optimal control problems are also RL methods [4]. The terminology in the two areas are analogue, for instance the term *learning* can be defined as "solving a dynamic programming problem without using an explicit mathematical model" [5]. Optimal control and dynamic programming terminology is adopted in this article.

We apply a RL algorithm to provide a solution to a multi-agent problem. More specifically, we address a problem in a simulated discrete-time warehouse setting with multiple robots/agents. The agents shall not collide with each other, and simultaneously complete tasks in the form of picking up boxes and delivering them to delivery stations. We define an episode as a single run of a simulation, from start to finish. The episode is finished after a fixed number of time-steps, or beforehand when all tasks are completed. With optimal control and dynamic programming terminology, the optimal control problem is characterized as deterministic, finite horizon, discrete time and multi-agent.

Optimal solutions to deterministic dynamic programming problems are not trivial and classic solutions suffer from the curse of dimensionality [4]. To solve the main problem, we consider a centralized process based on the idea of rollout and policy iteration proposed in [6]. The rollout algorithm is presented as a reliable method and relevant for on-line replanning [6], such as the main problem of this article. A variant of the chosen methodology has previously been applied on a multi-robot repair problem with partial observability [7], where results were presented with 2 agents. The problem of this article has full observability, however, we consider up to 10 agents with a considerable collision constraint between agents.

II. Preliminaries

A discrete-time dynamic system generates a sequence of states, containing the necessary information of the environment for optimization [6]. The states are influenced by a decision variable at each time-step called a control. At time step k, the state is denoted x_k and the control is denoted u_k. The state x_k is an element in a set of all possible states called the state space X. Likewise, u_k is an element in the control space U. If there is a maximum number of time-steps N, the system is called finite-horizon [5]. Mathematically, the finite-horizon deterministic dynamic system with a constant system

function can be expressed as,

$$x_{k+1} = f(x_k, u_k), \tag{1}$$

where $k = 0, 1, 2, \ldots, N-1$. The system function $f(x_k, u_k)$ determines the next state x_{k+1}, and is a function of the state and control solely and not any stochastic variable. Thus, $f(x_k, u_k)$ describes a deterministic system [5].

A policy is a set of functions that maps an agent's state to a control [4]. A policy can be expressed as,

$$\pi = \{\mu_0, \ldots, \mu_{N-1}\}, \tag{2}$$

where μ_k is a function that maps a given state into a control $u_k = \mu_k(x_k)$ [6].

A transition between two states given a control is associated with a cost determined by the function g_k. Thus, starting from state x_0, the total cost of a policy π is naturally stated as,

$$J_\pi(x_0) = g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k)), \tag{3}$$

where $g_N(x_N)$ is called the *terminal cost* [5].

III. Method

A. Environment

The environment is a discrete 14×14-grid in which each element represent one of the following objects, *free-space*, *wall*, *box*, *agent*, *agent-with-box*, and *delivery-point*. An agent object is distinct from another agent object to ensure multi-agent compatibility. This also applies to agent-with-box objects.

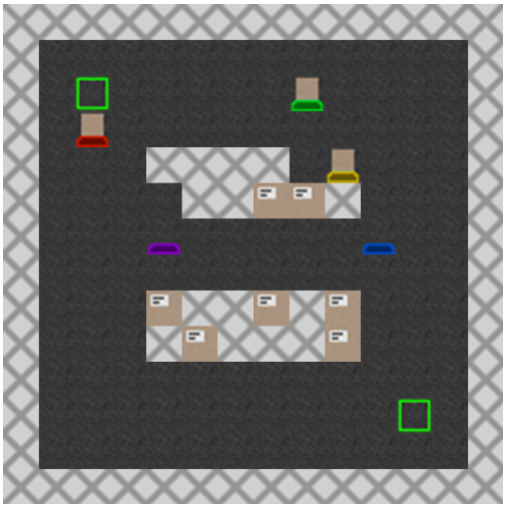


Fig. 1. Visual representation of environment with 5 agents and 10 boxes where 3 agents are currently carrying boxes.

As seen in Figure 1, the wall objects are placed as inner walls as well as an outer perimeter to enclose the environment. The inner walls represent warehouse shelves and accommodate boxes. In the areas designated as shelves, boxes are placed with randomly sampled positions at the beginning of an episode.

Each agent object has the attribute *target*, which holds the position of either a box, delivery-point, or a free-space object.

TABLE I
Costs per agent

State transition	Cost per Agent
Every	1
Picking up Box	−100
Deliver box	−1000

An agent may not have the same target box as another agent. Let B be a set containing all initial boxes,

$$B = \{b_1, b_2, \ldots, b_p\},$$

and let $T \subseteq B$ be the set of boxes that are currently targets of agents. Then, let the *vacant* boxes V be a set such that,

$$V = B \setminus T.$$

Let (y^i, z^i) be the position of agent i and (y_b, z_b) be the position of a vacant box b, then the target box of agent i is determined by,

$$b^i \in \arg\min_{b \in V} \Big\{ \mid y^i - y_b \mid + \mid z^i - z_b \mid \Big\}. \tag{4}$$

If there are several boxes with the same distance to the agent, the first one in the set is chosen.

At the beginning of an episode, each agent is assigned a target of a box according to the Equation (4), i.e the box with the shortest Manhattan distance. When an agent reaches its designated box, the agent and the box is transformed into an agent-with-box object. Additionally, the target of the agent is updated to a delivery-point, which of whom depends on the box's position. To relieve agent congestion, two differently positioned delivery points are present, as seen in Figure 1. If the box's initial position was in the upper shelf area, the agent's target is now the upper left delivery-point. On the other hand, if the box's initial position was in the lower shelf area, its target is then the lower right delivery-point.

When an agent-with-box reaches its target, i.e delivery point, it transforms into a regular agent again and the carried box vanishes. Again, the minimization problem in Equation (4) is solved after which the target is assigned to the agent. If there are no vacant boxes left, the agent is instead assigned a free-space object as target which is along the sides of the grid to leave room for the other agents to operate.

At every time-step, each agent has 5 different control options. Namely, either one step upwards, downwards, left, right or to stand still. An agent may not have the same position as a wall and may only have the same position as a box if the box is the target of the agent. If an agent chooses a control which contradicts with the rules above, its position does not change. An agent may choose control that results in a collision. A collision here is defined as two or more agents or agent-with-box objects occupy the same position in the grid.

The cost function g is characterized by the costs presented in Table I. A negative cost can be seen as a positive reward. In addition to the costs presented in Table I, there is a significant cost for collisions,

$$g_{coll} = 10^{12} \times 0.95^k.$$

The cost of collisions exponentially decreases with the number of time-steps k taken, ensuring that collisions early in the episode are more expensive. Also, if there are n number of agents colliding, the collision cost is multiplied with $n-1$. The multiplication factor implies that it is worse with a multitude of collisions rather than a collision between two agents.

The terminal cost $g_N(x_N)$ is assigned when transitioning to the final state N. If there are m agents and p initial boxes, the terminal cost can be expressed as,

$$g_N(x_N) = -100p - 1000p + mN.$$

The terminal cost ensures that, if the tasks are not completed, the total cost is positive.

At every time-step the information in the environment is summarized and represented by a state x [6]. In this case, the state consists of a matrix representation of the grid-space with integer elements representing various environment objects. In addition to the matrix, a state also carries a vector containing the targets of each agent.

B. Base policy

Let π, called the *base policy*, be a policy of form (2). The base policy used is a shortest path policy by the A-star algorithm [8]. That is, given a state x_k, the base policy outputs a mapping to control $u_k = \mu_k(x_k)$ such that it minimizes the following,

$$f_{A^*}(x_k, u_k) = g_{A^*}(x_k, u_k) + h_{A^*}(x_k, u_k),$$

where $g_{A^*}(x_k) = 1$ for all x_k and the heuristic h_{A_*} is the Manhattan distance between the agent and its target. Note that f_{A^*} above and f in (1) are different functions, which is also true for g_{A^*} and g_k in (3).

If the tasks are done in $n < N$ time-steps, i.e all boxes are delivered, μ_k maps every state to a control corresponding to a stand still for all $k \in [n+1, N]$.

C. Multi-agent rollout

Now consider m agents, where the control at time-step k is divided into m components corresponding to the choice of each respective agent,

$$u_k = (u_k^1, u_k^2, ..., u_k^m),$$

which belongs to the cartesian product of all individual control spaces,

$$u_k \in \prod_{l=1}^{m} U^l, \tag{5}$$

and U^l is the control space of agent l.

It is evident that the number of unique controls, i.e the cardinality of U, grows exponentially with the number of agents. In this case, where each agent has a constant choice of 5 controls, the number of controls in U is equal to 5^m, resulting in a computational cost of $\mathcal{O}(5^m)$ per time-step. Thus, using a standard rollout algorithm [6],

$$\tilde{u}_k \in \underset{u_k \in U}{\arg\min} \Big\{ g_k(x_k, u_k) + J_{k+1,\pi}(f(x_k, u_k)) \Big\}, \tag{6}$$

is not feasible for large number of agents.

Instead, a *one-agent-at-a-time* formulation of multi-agent rollout is used, which reduces the computation to grow linearly with the number of agents while still keeping a cost improvement over the base policy, see [6] for more details.

One-agent-at-a-time formulation trades control-complexity with state-complexity by adding $m-1$ intermediate states and corresponding cost-to-go functions J. These intermediate states are created by letting one agent at a time produce a new state after being assigned a control, hence the name. It is not until the last agent m, has been assigned its control that the official state transitions into a new one and produce a corresponding cost. Essentially, it transforms a single minimization problem in (6) into a sequence of m minimization problems,

$$\begin{aligned}
\tilde{\mu}_k^1(x_k) \in \underset{u_k^1 \in U^1}{\arg\min} \Big\{ & g_k\big(x_k, u_k^1, \mu_k^2(x_k), \ldots, \mu_k^m(x_k)\big) \\
& + J_{k+1,\pi}\Big(f\big(x_k, u_k^1, \mu_k^2(x_k), \ldots, \mu_k^m(x_k)\big)\Big)\Big\}, \\
\tilde{\mu}_k^2(x_k) \in \underset{u_k^2 \in U^2}{\arg\min} \Big\{ & g_k\big(x_k, \tilde{\mu}_k^1, u_k^2, \ldots, \mu_k^m(x_k)\big) \\
& + J_{k+1,\pi}\Big(f\big(x_k, \tilde{\mu}_k^1, u_k^2, \ldots, \mu_k^m(x_k)\big)\Big)\Big\}, \\
& \ldots \\
\tilde{\mu}_k^m(x_k) \in \underset{u_k^m \in U^m}{\arg\min} \Big\{ & g_k(x_k, \tilde{\mu}_k^1, \ldots, \mu_k^{m-1}(x_k), u_k^m)) \\
& + J_{k+1,\pi}\Big(f\big(x_k, \tilde{\mu}_k^1, \ldots, \mu_k^{m-1}(x_k), u_k^m\big)\Big)\Big\}.
\end{aligned} \tag{7}$$

Together, the components in (7) form a control $\tilde{\mu}_k(x_k) = (\tilde{\mu}_k^1(x_k), \ldots, \tilde{\mu}_k^m(x_k))$ which for all $k \in [1, N]$ generates the *rollout policy* $\tilde{\pi}$. In its essence, Equation (7) shows that each component uses the result from previous minimizations. In order, an agent tries a control while assuming all previous agents use their respective minimization result and all subsequent agents use their base policy. The cost-to-go functions J are calculated by simulation. This simulation is performed by creating a separate environment with state x_{k+2} and summing the resulting cost of using the base policy.

D. Policy modification and additional collision avoidance

An individual control u_k^l chosen at a state x_k for the agent by minimization l in (7) does not necessarily produce a unique minimum cost. There can be several controls that results in the same minimum cost. To choose one control in the set generated by the minimization, the controls that results in a position that is closest to the target is picked. However, if there are still controls with the same distances, a control is randomly sampled from the set.

Before every state transition, a one step simulation is performed to ensure a that the control is not resulting in a collision. If it does however detect that a collision is ensuing, the rollout procedure is recomputed with a randomly sampled

agent order. This shuffling and recomputation is done until the collision threat is alleviated or until this proceduere has been performed more than 80 times. A finite amount of random shuffling is done since a systematic iteration of all combinations is not feasible for large number of agents.

E. Simulations

The following selected subjects are studied in the context of this project,

1) Rate of episodic success, namely the percentage of successful episodes for different amounts of agents. An episode is deemed unsuccessful if agents collide or if the tasks are not completed within 200 time-steps.
2) The average amount of time-steps for different numbers of agents to complete the tasks in the same environment configuration.
3) Computational time for a time-step with 8 agents.

To provide the ratio of successful episodes, 100 episodes are performed with 10 randomly initialized boxes. At the end of every episode, the episode is determined successful if the resulting cost is non-positive. This procedure is then repeated for agent amounts ranging from 2 to 10.

In regards to the second group of tests, to ensure the number of time-steps between agent numbers is compared correctly, the same set of box configurations are used for these tests. These tests were executed with agent amounts ranging from 5 to 10.

The computational time is the average of 800 time-steps with 8 agents. It is computed using a computer with commercial-grade components.

F. Utilities

The project used Python as the main programming language as well as OpenAI gym for the environment. OpenAI Gym is a toolkit for RL problems [9] and the OpenAI Gym environment *gym-sokoban* [10] was used as inspiration and for rendering. The package *time* was used for measuring computational time.

IV. Results

TABLE II
SUCCESS RATE OF 100 EPISODES

No. of agents	Success rate
10	56%
9	64%
8	86%
7	93%
6	95%
5	94%
4	88%
3	78%
2	51%

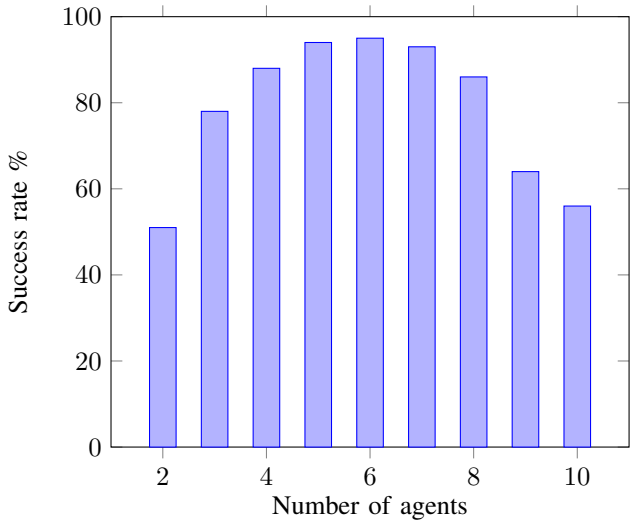


Fig. 2. Success rate percentage of 100 episodes. Specific values are found in Table II.

TABLE III
AVERAGE NUMBER OF STEPS, 10 DIFFERENT BOX CONFIGURATIONS

No. of agents	Average no. of steps
10	26
9	34
8	34
7	37
6	39
5	48

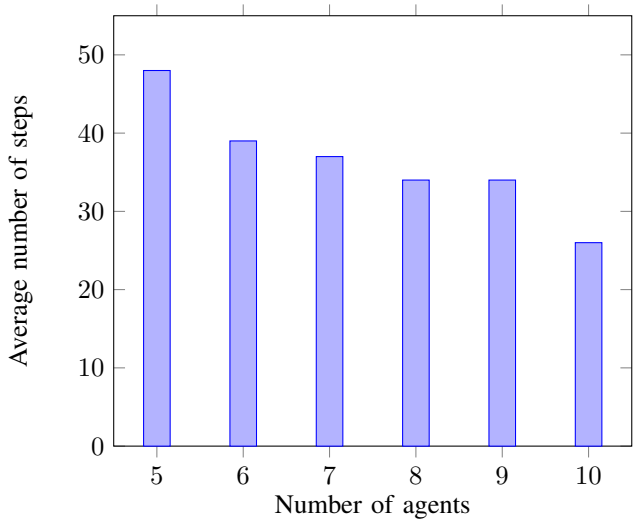


Fig. 3. Average number of steps for 10 different box configurations. This figure is a graphical representation of Table III.

TABLE IV
COMPUTATIONAL TIME PER TIME-STEP

No. of agents	Average computational time of a time-step [s]
8	1.6

V. Discussion

The results indicate that multi-agent rollout is a promising method for large-scale multi-agent systems with collision constraints. It is expected that the success rate decreases with increasing number of agents as difficult congestions occur more frequently, which is verified by the results in Figure 2. Similarly, the success rate for lower number of agents is also much lower than the maximum at 6 agents in Figure 2. This is on the other hand due to the agents not completing the tasks within 200 time-steps. This phenomenon is a direct result of the choice of costs, since it is possible to, by changing the costs, obtain a higher success rate for 2 and 3 agents. However, these changes negatively affect the success rate for larger number of agents. This does not indicate that multi-agent rollout is inappropriate for a smaller number of agents but that the method may require costs that are fine-tuned for the specific number of agents of the application.

It is important to note that the additional collision avoidance of randomizing agent rollout order was particularly effective for greater number of agents. Since a changed rollout order was helpful in accomplishing results in Figure 2, it can be compelling to investigate the topic of further optimizing rollout order for this problem. Optimization of rollout order is a subject discussed in [6] and implementation of it likely results in better performance but with a drawback in computation time.

As one can expect, the results in Figure 3 demonstrate that a greater number of agents complete tasks quicker, i.e. in less time-steps. However, with a increase in number of agents, a compromise in episodic success rate is made. These tests were not done for number of agents ranging from 2 to 4 as the tests require that the episodes are successful for all number of agents tested. Finding box configurations that lead to successful episodes for all agents requires a vast amount of search time, which was not deemed reasonable in this project.

Table IV presents a reasonable computation time for 8 agents, considering the chosen methodology involves on-line computing and re-planning. Although computation time varies drastically with the computational resources used, the numbers serve as reference for the speed one can obtain using a consumer level computer. Future work and a possible improvement of the computation time could be to implement deep neural networks as value or policy networks. The neural networks could be trained off-line to approximate the cost functions or the rollout policy through supervised learning. This could greatly reduce the on-line computational time with the downside of having a worse accuracy.

Another interesting idea to explore is that of dividing the environment into sections, known as *partitioning* [7]. This enables distribution of computation by parallelization between the different partitions. In this very environment, it is natural to think of the different shelves and their respective delivery points as partitions already. It is not difficult to imagine a warehouse with a large number of shelves and delivery points where this type of implementation could become practical.

Whereas this article addresses deterministic problems, real world warehouses are environments with stochastic properties. Whether it is human intervention or unforeseen technical problems, the agents must be able to identify the problem and adapt. Fortunately, it is fairly easy to extend the current formulation to allow for a non-deterministic environment as discussed in [6].

Additionally, a real world warehouse often have a perpetual workload in which the current finite-horizon formulation might be impractical. Again, an infinite-horizon version of the algorithm is discussed in [6].

VI. Conclusion

This project has considered the usage of a RL algorithm called multi-agent rollout to optimize a warehouse. The chosen algorithm allows for on-line replanning and is paired with a shortest path base policy. The results from the simulation indicates that the algorithm performs well considering the small computational time. However, a universal set of costs that works well for all agent numbers from 2 to 10, was not found. Thus, the methodology seems to work best when the number of agents is specified. Given the simplicity and promising results, multi-agent rollout applied to problems with collision constraints is worth further investigation.

Acknowledgment

The authors would like to thank their supervisor Yuchao Li for his invaluable guidance throughout the project.

References

[1] N. Vlassis, *A Concise Introduction to Multiagent Systems and Distributed Artificial Intelligence*. San Rafael: Morgan & Claypool Publishers, 2007, vol. 1, no. 1.

[2] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, 2008.

[3] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3826–3839, 2020.

[4] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed., ser. Adaptive Computation and Machine Learning series. Cambridge: MIT Press, 2018.

[5] D. P. Bertsekas, *Reinforcement Learning and Optimal Control*. Belmont: Athena Scientific, 2019, ch. 1.

[6] D. P. Bertsekas, "Multiagent reinforcement learning: Rollout and policy iteration," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 2, pp. 249–272, 2021.

[7] S. Bhattacharya, S. Badyal, T. Wheeler, S. Gil, and D. Bertsekas, "Reinforcement learning for POMDP: Partitioned rollout and policy iteration with application to autonomous sequential repair problems," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 3967–3974, 2020.

[8] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[9] (2021, Apr.) Getting started with gym. OpenAI, San Francisco, CA, USA. [Online]. Available: https://gym.openai.com/docs/

[10] M.-P. B. Schrader. (2018, Mar.) gym-sokoban. [Online]. Available: https://github.com/mpSchrader/gym-sokoban

Distributed Deep Reinforcement Learning for a Multi-Robot Warehouse System

Holger Stenberg and Johan Wahréus

***Abstract*—This project concerns optimizing the behavior of multiple dispatching robots in a virtual warehouse environment. Q-learning and deep Q-learning algorithms, two established methods in reinforcement learning, were used for this purpose. Simulations were run during the project, implementing and comparing different algorithms on environments with up to four robots. The efficiency of a given algorithm was assessed primarily by the number of packages it enabled the robots to deliver and how fast the solution converged. The simulation results revealed that a Q-learning algorithm could solve problems in environments with up to two active robots efficiently. To solve more complex problems in environments with more than two robots, deep Q-learning had to be implemented to avoid prolonged computations and excessive memory usage.**

***Sammanfattning*—Detta projekt handlar om att optimera rörelserna för ett flertal robotar i en virtuell miljö. Q-learning- och deep Q-learning-algoritmer, två väletablerade metoder inom maskininlärning, användes för detta. Under projektet utfördes simuleringar där de olika algoritmerna jämfördes i miljöer med upp till fyra robotar. En given algoritms prestanda bedömdes med avseende på hur många paket robotarna kunde leverera i miljön samt hur snabbt en lösning konvergerade. Resultaten visade att Q-learning kunde lösa problem i miljöer med upp 2 robotar effektivt. För större problem användes deep Q-learning för att undvika långvariga beräkningar och stor minnesåtgång.**

***Index Terms*—Deep-Q Learning, Multi-agent system, Neural network, Reinforcement Learning, Robots, Q-learning.**

Supervisors: Hamed Taghavian

TRITA number: TRITA-EECS-EX-2021:149

I. Introduction

Machine learning is a term that dates back to the 1950's, said to be coined by an American computer scientist called Arthur Samuel [1]. In machine learning, mathematical algorithms are used to analyze data in order to make predictions about the future.

Reinforcement learning is a subset of machine learning, that deals with algorithms adapting to a task through trial and error. While performing a given task, the algorithm will be rewarded for good behaviors and penalized for bad behaviors [2]. This form of machine learning will therefore lend itself well to solving problems in environments where all possible states and actions can be specified.

In this project, reinforcement learning was implemented to optimize the movements of virtual robots within a warehouse environment. The optimization problem consisted of enabling multiple robots to find optimal paths of transporting packages from one point to another. The robots were not allowed to collide with each other or with obstacles in the environment, forcing the robots to collaborate in the optimization process.

One of the purposes of placing automated robots in a warehouse environment is to relieve humans of tasks that are monotonous and/or physically straining. These types of robot systems are becoming more and more common in warehouses around the world. Out of Amazons 215 worldwide fulfillment centers, 26 of them are equipped with robotic systems aiding Amazon employees in packaging orders [3]. The performance of these robots must offset their investment cost, the robots need to be fast and they need to be reliable. As this report will show, reinforcement learning is one method that could potentially be useful in optimizing these robotic systems.

The reader of this report will in section II find preliminaries needed to understand the basic principles of reinforcement learning. In section III, the implementations in this project are explained in detail. The main results of the project are presented in section IV, which is followed by a discussion and a conclusion in sections V and VI respectively.

II. Preliminaries

A. Reinforcement learning

A reinforcement learning problem consists of an environment and one or several interacting agents. Agents are trained to find sequences of actions that maximize an accumulated reward. Reinforcement learning is illustrated in figure 1. Each possible action that an agent can perform corresponds to a reward that is given by a set of rules that are compiled into a *reward function* [4]. The environment state changes based on the actions taken by the agent. It is therefore possible to shape the behavior of the agent by establishing rules.

The reward function must be constructed so that an optimal action sequence, commonly known as an optimal policy, is developed over time. One way of inciting an agent to find the shortest path to a goal state is to punish the agent with a negative reward for each move it makes. Since the accumulated reward is being maximized, this would in theory force the agent to choose the shortest path possible to a goal state. Furthermore, the environment will also have to contain some terminal states. These are environment states in which a simulation is terminated. In the context of this project, terminal states includes states where an agent collides with an obstacle or another agent. A terminal condition will also be met if a maximum number of steps has been taken by the agent in the simulation, this is to avoid infinite loops within a simulation.

As the reinforcement learning agent seeks to optimize the accumulated reward, it is essential that the highest reward exactly implies the desired solution.

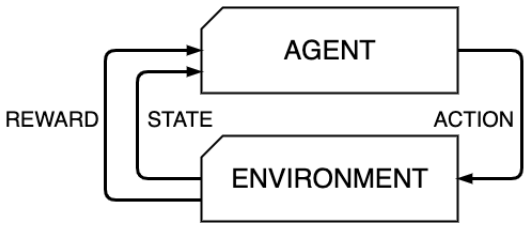


Fig. 1. The basic idea behind reinforcement learning.

At each step instance, the environment is in a state, which in the context of the project represents the robots positions in the warehouse. Each robot in this project is able to perform five different actions: 'up', 'down', 'left', 'right' and 'stay'. As seen in figure 1, the environment will update its state when an agent takes an action. In return, the agent receives a reward that is specified by the reward function [5].

B. Q-learning

One way of implementing reinforcement learning is to use a method called Q-learning. In Q-Learning, an agent calculates an optimal action sequence by evaluating which immediate action will be the most beneficial to the long term goal. An agent will be put in an environment where it will explore its surroundings while updating a Q-table according to the Bellman equation given in equation (1) [6]. The Q-table is a state-action table which contains numerical estimates of how valuable every possible action for a given state is.

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha(R(s_{t+1}) + \gamma \cdot \max_a Q_t(s_{t+1}, a) - Q_t(s_t, a_t)) \quad (1)$$

The elements that make up the Bellman equation are briefly explained below:

- s_t, state at time t. The environment will move within a set of discrete states.
- a_t, action at time t. The agent has a limited set of possible moves available to it. Each action will take the environment into a new state.
- α, learning rate. The learning rate determines how fast new information is accumulated by the reinforcement learning algorithm.
- $R(s_t)$, reward function. This function maps every state in the environment to a value.
- $\max_a Q_t(s_{t+1}, a)$. This function returns the action that will result in the highest Q-value at s_{t+1}.
- γ, discount factor. The discount factor weighs the importance of future rewards, a low-valued γ means that short term rewards are more important than long term rewards.

Choosing the most optimal action from the Q-table is called exploitation. Since it is established that an agent learns from past experiences, it might develop a sub-optimal policy if it is not exploring the environment enough. If an agent has not experienced a state in the past, it has no way of knowing if it is good or not. The exploration/exploitation dilemma is about choosing between actions that have previously yielded positive outcomes and actions for which the outcomes are not yet known, but in the end might yield a greater accumulated reward.

The epsilon-greedy algorithm [7] is used to balance the exploration and exploitation probabilities of an agent over time. The exploration rate starts high and is decreasing over the course of a simulation. Agents using the epsilon-greedy algorithm take actions either by random when exploring or by looking up the optimal action in the Q-table when exploiting. The statistical probability of an agent choosing one over the other can be set by using the epsilon parameter. The pseudo-code for the epsilon-greedy algorithm is given in Algorithm 1.

Algorithm 1 Epsilon-Greedy

```
Set an epsilon value between 0 and 1.
Generate a random value between 0 and 1.
if random number > epsilon then
    Choose exploitation
else
    Choose exploration
```

C. Neural Networks and Deep Q-Learning (DQN)

A Q-table for n robots taking a possible actions, moving in an $m \times m$-sized environment would have

$$(m^2 a)^n$$

elements. This means that if four robots were put in a 10×10-squared environment, the corresponding Q-table would have 62.5 billion elements. This shows that using Q-learning in more complex scenarios will certainly become computationally problematic. Using neural networks is way of overcoming this problem. In deep Q-learning, neural network replaces the Q-table that is used in Q-learning [8]. The majority of the neural network maintains a constant memory size no matter the number of input and outputs. Only the input layer and output layer changes in size.

A neural network is made up of layers of nodes. Figure 2 illustrates the basic structure of a neural network with fully connected layers. Having fully connected layers means that there is a connection between each node in one layer to the adjacent layers' nodes. Each node has a weight and an activation function. The weights in each node are calibrated so that the network produces a wanted output for a given input.

When training a neural network, data is supplied to the network with inputs for which there are known outputs. The weights of each node are calibrated to minimize the error between the desired output and the actual output. As reinforcement learning relies on past experiences to improve, certain metrics of state-action pairs needs to be stored as memories during a simulation. These memories will then be used by the neural network to adjust its weights to predict future events with high precision.

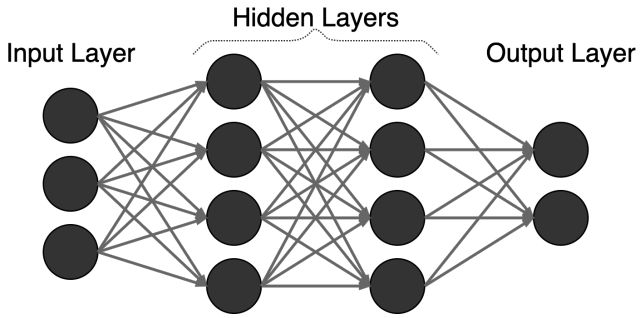


Fig. 2. The basic idea structure of a neural network.

III. Implementation

A. Program structure

When designing the simulation software, the project group focused on building a modular system. This was motivated by the fact that some parts of the program would remain the same regardless of what specific learning algorithm was implemented. For example, the self developed environment module was used in combination of different algorithms. Figure 3 shows a top level diagram of the implemented program structure.

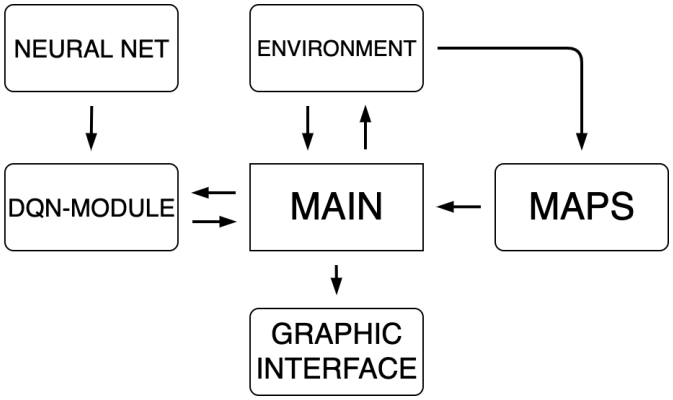


Fig. 3. Top level diagram of the simulation program structure.

B. Environment design

The warehouse environment used in the simulations was a self developed python class implemented using NumPy arrays in Python [9], [10]. Python is a well known programming language that the project group saw fitting for the project. Multiple default warehouse environments were designed, varying in size, robot count, obstacle placements and in robot drop-off/pickup points.

The reward functions were constructed so that continuous exploration and exploitation would eventually lead to the goal states being found and prioritized. Colliding with an obstacle would result in a large negative reward and reaching the target position would result in an equally large but positive reward. The reward values that were used during simulations with Q-learning algorithms are given in table I.

TABLE I
Reward values used for Q-learning simulations.

Event	Reward
Collision with an agent or obstacle	-1
Step	-0.1
Reaching target position	1

A larger environment implies more possible choices for the agents. Longer sequences of actions will have to be executed to get to a target state. Because of this, additional measures were implemented to encourage the agents' policies to converge in a reasonable time. During later stage experiments, the agents were given weakly suggested paths at the initiation of a simulation. If these paths were being followed, the agents would receive additional rewards. The additional rewards were set relatively low, since the weakly suggested paths were merely educated guesses. It was important that these suggested paths were not affecting the final outcome too much. The project group refers to this method as *path guidance*. The reward values that were used during simulations with deep Q-learning algorithms are given in table II.

TABLE II
Reward values used for deep Q-learning simulations.

Event	Reward
Collision with an agent or obstacle	-1
Step	-0.1
Following suggested path (optional)	0.1
Reaching package	2
Delivered package	5

C. Centralized Q-learning

The first step in the project was to implement a centralized (single agent) Q-learning model for 1-2 robots. Figure 4 shows an example of a simple environment that was initiated for a single robot. The current position of the robot is represented by a lowercase a, the goal state is represented by an uppercase A. H represents an obstacle. This symbol convention was used throughout all simulations in the project. For example, if three robots were used, they would be represented as 'a','b' and 'c'.

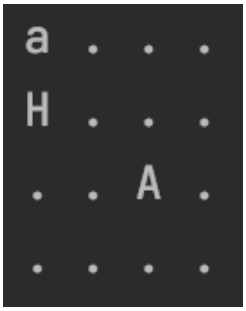


Fig. 4. An initiated warehouse environment for a single agent.

The algorithm used for the Q-learning implementation is given in pseudo-code in Algorithm 2. A more detailed of this algorithm can be be found in [11].

Algorithm 2 Centralized Q-learning for 1-2 agents

```
Create environment, incl. robots and obstacles
Create Q-table, set all entries to zero
for episode = 1, 2, ..., EpisodeMax do
    Reset environment.
    for step = 1, 2, ..., StepMax do
        Choose exploration or exploitation
        if exploration then
            Choose a random action
        end if
        if exploitation then
            Q-table lookup for optimal action
        end if
        Update position, reward and Q-table.
        if step=StepMax or collision=True then
            Break
        end if
    end for
    Decrease exploration rate
end for
```

The exploration rate would decrease by a small amount for every simulation episode. By letting the exploration rate decrease over time, the robot would depend more and more on the Q-table as the simulation went on. In theory, a decreasing exploration rate would eventually lead to the agent policy being optimized to take the shortest path to the goal state. A convergence in acquired reward would indicate reaching such an optimal policy.

D. Deep Q-learning

The deep Q-learning algorithm, DQN-module, was implemented with the aid of the Python packages $TensorFlow$ and $Keras$ [12]. These open source packages offered neural network construction tools deemed fit for this particular project.

A fully connected sequential network with three hidden layers and 16 nodes per layer was implemented. The ReLu (rectified linear unit) activation function was used for all nodes. Due to the believed non-linearity of the task, multiple layers were used. A reasonably small network size was chosen to speed up the weight calibration process. The chosen optimizer was ADAM [13], which uses the MSE (mean squared error) network loss function.

To have a distributed system, a separate neural network was considered for each agent in the DQN-module. We assumed that the neural network in each agent can access the other agents' coordinates and current tasks. A task was being defined as "picking up a package" or "delivering a package". These states were given to the neural networks to enable enhanced decision-making. The pseudo-code for the algorithm used for distributed decentralized (multiple agent) deep-Q learning can be found below in algorithm 3.

In this project, the following data was stored for each simulation step:

- The current state of the environment.
- The next state of the environment after all agents have made a move.

Algorithm 3 Distributed deep-Q learning for multiple agents

```
Create environment, incl. robots and obstacles
Initiate one neural network per agent, set gamma and
epsilon parameters for networks.
for episode = 1, 2, ..., EpisodeMax do
    Reset environment and metrics.
    for step = 1, 2, ..., StepMax do
        Collect agent action requests.
        retrieve updates from environment.
        if selected step was reccommended then
            give additional reward
        end if
        store data collected for training
        let networks replay stored data
    end for
end for
```

- The action that the agent just executed.
- The reward for taking the action.
- If the agent landed in a terminal state or not.

This data tuple was stored in a memory bank of the last 6000 episodes. After each time the simulation terminated or if 20 steps had been executed in the simulation episode, the neural network replayed a batch of four randomly selected memories out of the memory bank. This was implemented so that the neural network had the chance to re-calibrate its weights frequently. The replay procedure algorithm is described in Algorithm 4.

Algorithm 4 Replay memory procedure

```
Sample batch of memories (4).
for each batch of sampled memory do
    Set target = reward.
    if agent did not land in a termination state then
        target = reward + γ · max(output nodes' values).
    end if
    Let the neural network predict output values.
    Adjust network weights to match the target output.
end for
```

Note that parameter γ used in the training sequence will in a sense play the same role as the discount factor in Q-learning, and will therefore affect the behavior or how the neural network adapts to the environment.

IV. Results and Analysis

A. Q-learning

Figure 5 shows the average success rate per 200 episodes of a single robot placed in an environment that is shown in figure 4. A successful episode is defined as the event in which a robot moves from its starting position to its target position without colliding with an obstacle. In this simulation, a Q-learning algorithm in combination with the epsilon-greedy approach was used.

When using a centralized Q-learning model with two robots, their starting positions and target positions were placed in

the environment as in figure 6 so that the robots would have crossing paths. This was done to force collaboration between the robots. The result of convergence can be found in figure 7.

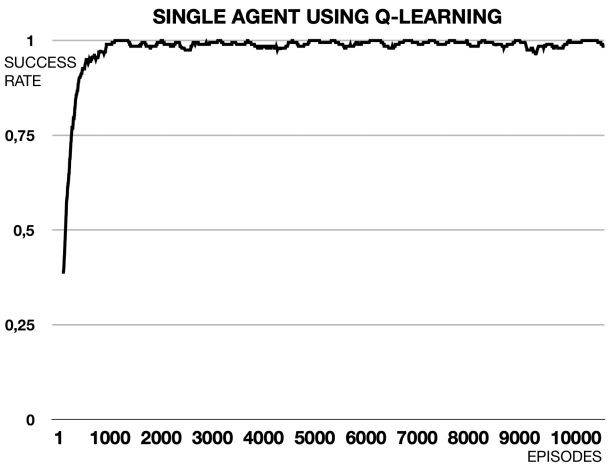


Fig. 5. Q-learning for a single robot. γ = 0.98, α = 0.9. 200 episode moving average.

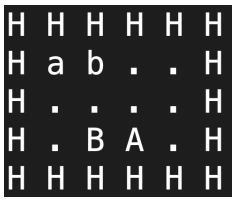


Fig. 6. Environment used in decentralized DQN simulation and Q-learning.

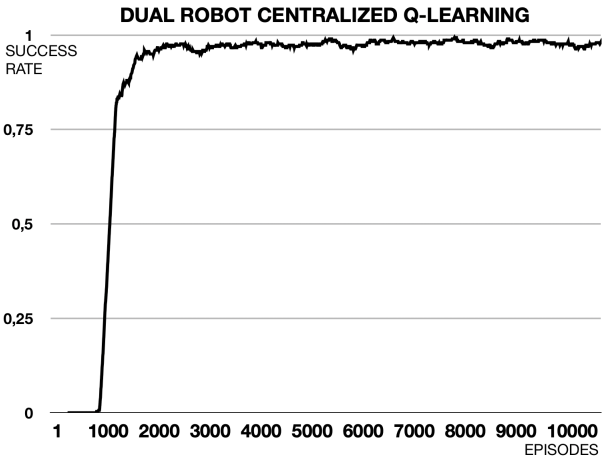


Fig. 7. Q-learning for two robots. γ = 0.98, α = 0.9. 300 episode moving average.

The Q-learning algorithms performed as expected. Converging policies can be seen in both figure 5 and in figure 7. The data gave a clear indication that an agent was in fact able to use data from the environment to improve it's behavior. It was observed that the case of having multiple robots required more episodes to converge. The data gathered from these simple experiments were taken into consideration when taking a step further with deep Q-learning.

B. Deep Q-learning

Deep Q-learning algorithms were applied to the same environment setup as in the Q-learning dual robot simulation, see figure 6. Instead of defining success as reaching their target positions, the robots now had to retrieve packages at their target positions and then return to their starting positions. This modification was made to fully comply with the project instructions. Figure 8 shows the results of a simulation where a decentralized deep Q-learning algorithm was used.

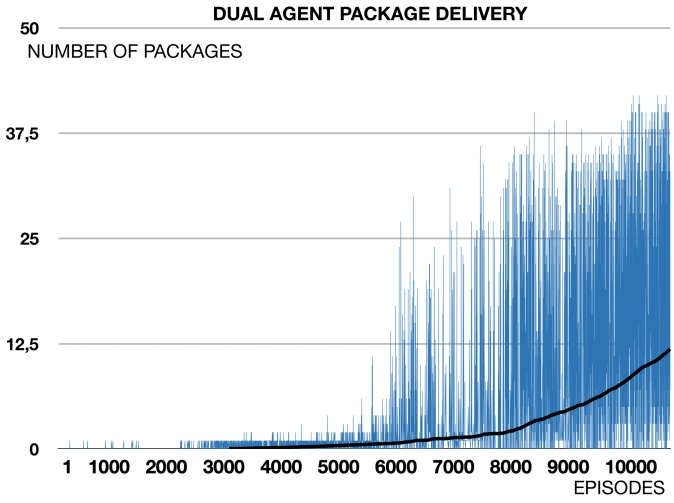


Fig. 8. Deep Q-learning for two robots. γ = 0.99, learning rate = 0.0001, 3500 episode moving average.

In figure 9, a higher learning rate in combination with path guidance were used to see if a faster convergence could be achieved. The data shows that the extra reward function in combination with high learning rate greatly increase convergence rates. Figure 9 indicates convergence after around 1500 episodes, while a convergence is not observed in figure 8 even after 10 000 episodes.

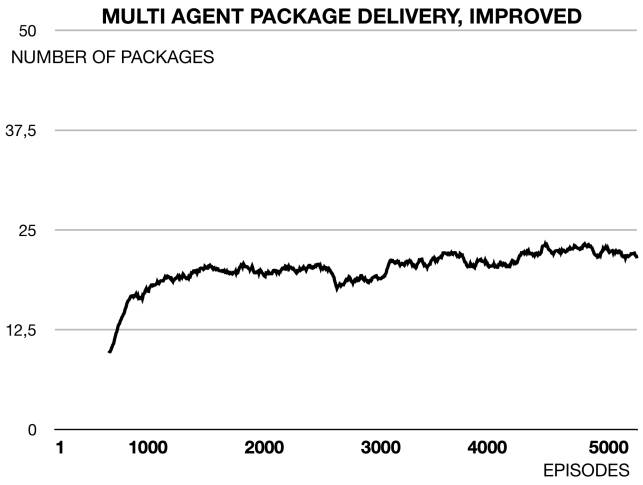


Fig. 9. Deep Q-learning for two robots. γ = 0.7, learning rate = 0.001, 400 episode moving average.

Figure 10 shows an environment in which a simulation with four agents was run, this simulation was made to verify that the improved DQN solution was scalable. In figure 11, results show that the algorithm is reaching an upper limit of 118 total packages delivered within 300 steps for each episode in increased frequency over time. By the looks of the graph, this could be the theoretical limit of package delivery.

```
H H H H H H H H H H
H . . . H . A . . H
H . a . H . . . . H
H . . . . . . B . H
H . b . . . . H H H
H . . . . . . C . H
H . c . . D . H H H
H . . d H . . H . H
H . . . H . . H . H
H H H H H H H H H H
```

Fig. 10. Environment used in decentralized DQN simulation and Q-learning.

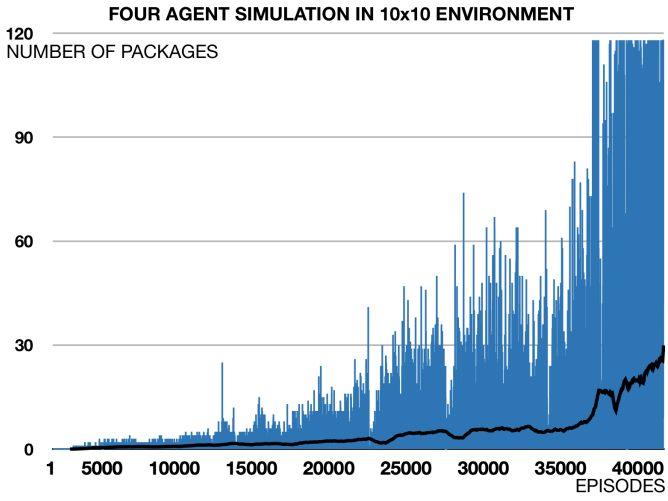


Fig. 11. Deep Q-learning for 4 robots. γ = 0.65, learning rate = 0.001, mimimum exploration rate = 4%, 1200 episode moving average.

V. Discussion

A. Q-learning

Simulations revealed that centralized Q-learning was an effective algorithm when applied to problems with 1-2 robots. Figure 7 reveals that the policy convergence is slower in the case with multiple robots than with a single robot. The explanation to this phenomenon is the number of possible actions available to the algorithm has been increased exponentially, compared with the previous algorithm. This results in a much larger Q-table, which increases computation complexity.

The moving averages in figures 5 and 7 are statistically unlikely to reach one since the exploration rate of the robots never reaches zero. A non-zero exploration rate implies that there still is a possibility for the robot make a random move that results in a collision. The non-zero exploration rate was kept to mitigate the risk of a robot getting stuck in a sub-optimal policy.

The exploration rate decay choice in the epsilon greedy algorithm proved to affect convergence speed greatly. When the exploration rate decay was set to a high value, much greater convergence rates were observed. However, if the parameter was set too high, the lack of exploration lead to a bad policy development as the agent would not find the goal state.

B. Deep Q-learning

When plotting the moving average of 3500 episodes, a near exponential point curve can be viewed in figure 8. Due to prolonged computation time, the simulation was terminated after 10000 episodes. Although the data is insufficient to display a converging policy, a clear positive trajectory can be observed.

During the initiation of this project, the project group set a goal of being able to optimize the movements of four robots in an environment grid of size 10 by 10. It can be observed in figure 11 that the decentralized deep Q-learning algorithm that was implemented succeeded with the set goal. The results from both the Q-learning algorithm implementation and the simpler dual agent scenarios gave great insight of what could speed up the process of converging to an optimal policy in a more complex environment.

A few interesting observations were made during the simulation of the 10x10 environment. The exploration rate was capped at 4% to enable continuous optimization improvement, and it was clear from the developed agent policies that certain robot movement patterns were adapted so that the robots would coordinate to avoid collision. Another interesting fact is that even though all agents had the same neural network structure with the same input states and output states, policies were different in the end. The initial positions were actually affecting the neural network. At last, even though all the agents where maximizing their own reward, collaboration seemed to be the winning strategy for all agents.

The project group also ran simulations where the robots had no information about the other robots positions or trajectories. As in previous scenarios, robot paths had to be crossed to succeed with the given tasks. What became clear was that the lack of information made some agents aggressive and some agents very passive. We suggest that the reason for this policy development was that the aggressive agent found a higher long term reward policy that would sometimes lead to collisions. The passive robots did probably not find such policy and would therefore optimize its policy to not collide with any obstacle, including other robots. It is believed that if the agents do know each others states, they can take the other robots states into consideration when reaching an optimal policy.

C. Path guidance

One method that was implemented in the project to improve solution convergence in deep Q-learning was to give the robots weakly suggested paths to their targets. The suggested path for a given robot did not take into account the paths of the remaining robots, it was left to the neural networks to

determine how the robots would handle meeting points. This method was applied to the same problem that was used to simulate the decentralized deep Q-learning algorithm and the results are depicted in figure 9. Comparing this figure with figure 8, it can be deduced that path guidance results in a faster convergence. This proved, as expected, that it was beneficial to model the reward structure such that the agent did not spend excessive time on exploiting and exploring areas of no interest.

Implementing path guidance is by the authors seen as one of the greater successes in this project. This method enabled the algorithms to converge faster towards an optimal solution. Reducing as many unnecessary computations as possible plays a crucial part in making these algorithms work in larger environments with more robots. Moreover, this result shows how important reward-shaping is when developing reinforcement learning algorithms.

D. Modeling the reward function

A key task in the project was to design an effective reward function. It became apparent during early simulations in the project that an ineffective reward function might result in situations where the robots never reach their goal states. If a goal state is not found by agent exploration, the agent would still try to maximize reward. Avoiding as much negative reward as possible until an episode terminated would be an option that traps an agent in a local maximum rather than a global maximum. This behavior highlights one of the biggest challenges in reinforcement learning. It must be ensured that the global maximum is found. The idea of local and global maxima is illustrated in figure 12.

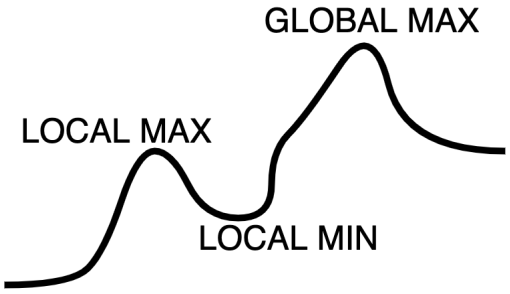


Fig. 12. Illustration of global and local maxima and mimima.

Similarities can be drawn between reinforcement learning and biological evolution. It is common knowledge that organisms evolve to better adapt to their surroundings. Every mutation in an organism must be beneficial to that organism in terms of survival and/or replication. If each mutation is not an improvement, it gets abolished by nature. The same principle applies to reinforcement learning. Since reinforcement learning always optimizes an accumulated reward, agents are not encouraged to explore if it does not come with any long or short term benefits.

A problem that arose regularly during the algorithm development was in fact that robots would move back and forth between two states repeatedly because it was most likely stuck in a local maximum. In figure 13, an environment is shown where a robot would statistically have trouble finding the goal state without any suggested paths, like the ones mentioned in III-B. The statistical likelihood of reaching the goal state in the case of a completely randomized policy is given in (2). Even though the chance to find the end state randomly is not zero, it is still very unlikely. In the context of this project, it shows for example that the epsilon-greedy algorithm might not be sufficient on its own in every scenario.

$$\frac{1}{(\text{possible actions})^{(\text{correct steps needed})}} = \frac{1}{5^{10}}. \qquad (2)$$

```
H H H H H H H
H . . . H A H
H . H . H . H
H a H . . . H
H H H H H H H
```

Fig. 13. An example of a difficult environment.

E. Optimizing the system

As mentioned in [14], optimizing neural networks is typically a very complex task that is often time consuming and requires a lot of computational resources. What in part makes this task difficult is the number of parameters and design choices that needs to be taken into account. What values should the discount rate be set to? Should these parameters be constant or should they change between simulations? Does the reward function give the agents incentive to move towards the goal state? How many nodes and layers should a neural network in a deep Q network consist of? These are some of the questions that faces a designer of a reinforcement learning system. The authors of this report struggled to find any clear correlations between different parameters.

In this project, all simulations were run on a personal computer with a 2,9 GHz Dual-Core processor and a 8 GB 2133 MHz RAM module. It took approximately 12 hours to compute the simulation for which the results are shown in figure 8, and 40 hours for the results in 11. Even though the simulation was run on a personal computer, it was still a simulation on a very basic problem. This hints at how much computer power would be needed for problems more closely connected to the real world.

F. Real world applications

This project was centered around a simplified model of the world where the environment was modeled as a relatively small grid of possible robot positions and actions. In order to use the algorithms that were developed in this project in a real environment, certain requirements would have to be put on the environment layout.

One idea is to create a warehouse layout that is based on robots being in discrete positions. If the robots were to move on a grid of rails for example, the environment would

closely resemble the simulated environments in this project. Each position in the grid could contain a certain product which a robot collects as their task. If a warehouse like this was built, robots could use reinforcement learning to optimize movement patterns. Separate control systems could then be used to ensure that the robots are being correctly placed in valid grid locations. Before the robots are even placed in the real environment, logged data from previous customer orders could be used to train the algorithm to perform well in simulation. With the use of large amount of computer power, every kind of scenario that the robots might be experiencing in the real world would be already experienced, and optimized, in the simulation.

G. Future Work

- A clear strategy for tuning the parameters used in the algorithms remains elusive. Further investigation into how the system parameters affects performance would likely be worth spending more time on.
- To further close the gap between simulation and reality, more states for the robots can be implemented to train the algorithms on more realistic scenarios. Battery capacity and storing capacity could be two such states that could affect robot behavior in different ways.
- In the path guide implementation, each step in the weakly suggested path would have to be given manually by the algorithm designer. Since this method showed promising results, automating this step by for example using Dijkstra's algorithm [15] to find a suggested path would be an idea worth exploring.

VI. Conclusion

In this project, we have demonstrated the effectiveness of reinforcement learning algorithms in optimization of robots movements within a virtual warehouse environment, through simulations.

The machine learning algorithms that were explored in this project were Q-learning and deep Q-learning. Q-learning was a sufficient algorithm when applied to 1-2 robots while deep Q-learning was necessary for solving more complex problems. It was shown that the implemented deep-Q learning algorithm used for policy optimization could be scaled up.

Shaping the reward functions turned out to be one of the key tasks in implementing effective algorithms. The project group experimented with a method that presented the robots with weakly suggested paths during training initiation. This method proved to be successful and enabled the robots to find an optimal policy faster.

Acknowledgment

The authors would like to thank mentor Hamed Taghavian for his most valueable feedback and collaboration throughout the project.

References

[1] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959.
[2] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, Massachusetts: The MIT Press, 2014, ch. 1.1.
[3] (2021, Apr.) What robots do (and don't do) at amazon fulfilment centres. [Online]. Available: https://www.aboutamazon.co.uk/amazon-fulfilment/what-robots-do-and-dont-do-at-amazon-fulfilment-centres
[4] P. Dayan and C. J. Watkins, "Technical Note, Q-Learning," *Machine Learning*, vol. 8, p. 280, 1992.
[5] (2021, Apr.) Introduction to rl and deep q networks. [Online]. Available: https://www.tensorflow.org/agents/tutorials/0_intro_rl
[6] M. Wiering and M. van Otterlo, *Reinforcement Learning: State-of-the-Art*. Berlin: Springer Science Business Media, Mar. 2012, ch. 1.6.1.1.
[7] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, Massachusetts: The MIT Press, 2014, ch. 2.2.
[8] (2021, Jan.) Reinforcement learning lecture by Emma Brunskill: CNNs and Deep Q learning. [Online]. Available: https://web.stanford.edu/class/cs234/slides/lecture6.pdf
[9] (2021, Apr.) Numpy. [Online]. Available: https://numpy.org//
[10] (2021, Apr.) Python. [Online]. Available: https://www.python.org/
[11] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, Massachusetts: The MIT Press, 2014, ch. 7.6.
[12] (2021, Apr.) Keras. [Online]. Available: https://keras.io/
[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Conf. ICLR: 3rd International Conference for Learning Representations'15*, San Diego, USA, Jan. 2015.
[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, Massachusetts: MIT Press, 2016, ch. 8.
[15] K. Eriksson and H. Gavel, *Diskret Matematik - Fördjupning*. Malmö: Holmbergs i Malmö AB, 2012, ch. 8.2.1.

CONTEXT C – PART II
LEARNING IN DYNAMICAL SYSTEMS

POPULAR DESCRIPTION

Will Your Next Coworker be a Robot?

The fourth industrial revolution is currently thriving. In order to solve the large-scale problems of tomorrow, we need to improve the abilities of the machines. Abilities to recognize complicated patterns and find good problem-solving strategies are important for making appropriate decisions. These pattern recognition techniques have been proven to be helpful tools to offload mental work.

We, as humans, have an innate ability to see patterns in our environment and from those patterns we are able to make proper decisions (i.e actions), to achieve a goal such as drinking a glass of water when thirsty. This might seem easy enough for us, but for computers and machines, this is a field that has only recently shown some success in smaller specific tasks. The future might hold advancement to solve more general and complex problems.

Learning by doing is an ability we observe in ourselves and is the basis of our decision-making process. Can we make computers do the same? If so, how? One way to achieve this is to assign values to actions, corresponding to how "good" the action is, and let the computer pick the action with the highest value, which coincides with the best action. However, we could not possibly define every action. If you think about it, there are millions, billions, even trillions of actions to take. Instead, the idea is to let the computer learn to value actions itself through trial and error, and reinforce good actions likewise inhibit bad actions. Building up an intuition based on its previous experiences. This process is similar to how animals, like dogs, learn.

The achievements of today should be viewed as a stepping stone towards a future of intelligent machines, where smart machines do our chores, and where smart machines help companies with better decision-making for a brighter future. Even your next coworker could be a helpful robot, so who knows what the future might hold?

SUMMARY OF PROJECT RESULTS

With the rapid evolution of computers and the ever increasing computing speed, more and more advanced learning algorithms can be implemented and explored for various tasks. These learning algorithms aim to mimic nature's way of learning, but in a simulated environment, and the implementations mostly contain pattern recognition. The applications for learning algorithms vary and include, for example, playing games, or portfolio optimization where an algorithm is taught to optimize certain factors of stock portfolios given numerous variables. One of the most active research areas that implement learning algorithms is in autonomous vehicles.

The project group C4a opts for an empirical approach to the reverse the engineering problem of an optimal portfolio. Today, there is a consensus among researchers that Markowitz mean-variance theory, which is a theory for finding an optimal portfolio based on a set of risk parameters and return predictions, is ill-suited to be applied to real-world scenarios. By instead analyzing empirical data from an actively managed mutual fund, we can draw conclusions regarding the extent to which technical indicators are used in trading. Findings indicate that sole technical input is insufficient for predicting subsequent trading action. Using similar machine learning methods, the trading strategy of the fund is optimized using the same technical indicators to improve the timing of trades. Further studies could include more input parameters in the search for technical trading predictors, and more intricate models for optimizing the trading performance could yield even greater returns.

The project group C4b has formulated an optimization problem based on Markowitz thesis of modern portfolio theory. Basically, the idea is to minimize the risk for a given level of expected return. The formulated model will include different parameters such as the expected return, the variance (risk), and a risk aversion parameter. With historical data from existing

funds different parameters can be estimated and from there we can derive a method that reverse engineer the formulated optimization problem, thus attempting to find the parameter of the risk aversion.

In future projects, further investigations regarding more sophisticated portfolio optimization could be implemented. Furthermore, it is logical to assume that portfolios focusing on different sectors would base their trading algorithms on different theories. Hence, a broader scope of this investigation would be to potentially find similarities/differences across different sectors. Finally, the proposed method is implemented by assuming efficient returns on the portfolios over an X year period. This has been proven to be inaccurate by various researchers.

The groups in project C5 primarily focus on implementing deep reinforcement learning algorithms and decision-making processes. However, the main objective differs slightly between group C5a and C5b.

In project group C5a we solve two different environments from OpenAI (ContinuousLunarLander and CartPole), using different deep reinforcement learning algorithms (DQN and DDPG). In CartPole the task is to balance a pole stuck to a cart, and for LunarLander the task is to land a moonlander. We then put our main focus on the production and detection of adversarial attacks on these trained network policies. Adversarial attacks are designed to make small perturbations to the input data in such a way that the model makes the wrong choice. The adversarial attacks are performed using the FGSM-algorithm, which is a method that tries to optimize the effect that the perturbed data has on the performance while keeping the total change of the data under a certain threshold. We then check its efficacy on both the policy used to produce them as well as how it transfers to other policies trained to solve the same environment. Lastly, we also look at ways of detecting adversarial attacks on network policies by training a binary classifier, meaning a classifier that categorizes data into two groups (in our case, under attack or not under attack), to distinguish perturbed examples from normal ones.

In project group C5b our main intent is to compare three different reinforcement learning (RL) algorithms using a simulated environment of the popular physical game tag with two acting agents and an arbitrary amount of obstacles. The two agents have opposite purposes where they receive rewards primarily based on the distance between them. We compare Q-learning, Deep Q-learning (DQN), and Double Deep Q-learning (DDQN) in order to find which training method is most optimal for playing tag. We consider the most optimal algorithm to be the one that takes the least amount of time to train the agents to fulfill their purposes.

In future projects, concerning both C5a and C5b, more focus could lie within the robustness of the RL algorithms. Currently, in most implementations, a small disturbance in the input may result in an amplified disturbance in the output, resulting in most cases the wrong results. If an algorithm is robust, it is less vulnerable to disturbances. This is one important aspect that needs further investigation.

IMPACT ON SOCIETY AND ENVIRONMENT

Artificial intelligence (AI) is a widely discussed research field when it comes to, even the most basic, ethical questions. Who is to be blamed in situations where an AI favours one particular group of people? How big of an impact does AI have on the environment? These are just a few examples of many discussed questions that concern AI.

Responsibility is a big ethical question in the AI science field. Who would be responsible for an action of AI? A common ethical conundrum is self-driving cars which may have to decide who to run over, in extreme cases. We argue that in these cases the companies that approve the products, are the ones that are responsible for the decision-making of an AI that is implemented in, for example, self-driving cars. It is them who have the control over product releases and should verify that their product does not contain any unwanted characteristics, thus any existing attribute is considered to be wanted by the company.

There are also issues with producing AI that acts ethically. It is very easy to cause an AI to be misaligned with our values, even when the engineers had no intention to create an unethical AI. It turns out it is extremely difficult to make a framework for our ethical codes, and more so, prove that the AI always follows it. Microsoft made a chatbot named Tay, and let it loose on Twitter, but unfortunetly had to be shut down shortly after its lauch, becuase the chatbot was writing Nazi progagande and

racist slurs. Twitter also utilizes an automatic cropping tool on user's photos, which consistently croups out colored people if a white person is present, even when the subject is a known figure such as Obama. We argue it is important that AI does not incorporate society's prejudices and unfavorably acts against a certain kind of group.

Another issue of artificial intelligence is the question of privacy. The quintessential task for an AI-model is to search for patterns in the data. Consequently, for the model to be able to make predictions about an individual it is necessary to put in data corresponding to the behaviour of that individual. This has led to large amounts of personalized data being collected, and said collection is not necessarily consensual. It might be reasonable to be able to shop for e.g groceries or shoelaces without being tracked by a company that wants to sell you, or other customers, more products. Following many controversial discoveries in regards to the extent private companies collect personal data and the consequential usage of it, some argue that regulatory actions are needed.

In machine learning, models are said to be fair or to have fairness if results and/or decisions are independent of sensitive data such as race, religious beliefs or ethnicity in occasions when these attributes should have no impact. Meaning that these traits should not correlate with the computer made choices a model produces. This is to a large extent a problem of making sure that you are not including data that you do not want the model to draw conclusions from.

One other important aspect about machine learning algorithms (and maybe especially neural networks) is the question of explainability. Neural networks are something that is often treated as a black box, since it is very hard to understand what exactly a network is doing due to the very large number of parameters. This could potentially be a problem as we as a society start relying more and more on AI in most areas of life. Imagine for example being judged for a crime by an AI system without being given any explanation of what you did wrong, or a doctor using AI to predict cancer in patients, while not being able to understand at all which predictors the model based its prediction on.

Sustainability is currently a major topic in today's politics. Many new scientific breakthroughs may, directly or indirectly, tackle the issues of our unsustainable lifestyle. Machine learning is no different. It will lead to a healthier and more sustainable world. Sustainability is often divided into its three principles: Environmental, Society & Economic, each of the aforementioned will be discussed below.

The environment benefits largely from the emergence of sophisticated artificial intelligence, learning from large data samples of how to efficiently produce crops, using satellite data to anticipate and prevent wildfires, or simply to calculate the lowest energy consumption route for product transportation are examples of environmental benefits which are directly supported by various AIs.

Artificial intelligence is already being used in most aspects of life today, which allows everyday citizens to finish their tasks more efficiently and therefore enables them to live a more stress free life.

With a growing population and more consumption there is a big pressure on our common resources. A solution to create a more sustainable setting is to take help of artificial intelligence and to use it to create smart solutions and to get circular economics.

In conclusion, we argue that in most cases it is the distributor, usually companies, of a product that is ethically responsible for its product, since the distributor has control over the production. If the distributor decides to launch a product he may potentially cause harm to certain people or the environment, but he also has the power to decide not to, which may cause less harm.

Machine Learning Methods for Predicting Trading Behaviour of an Actively Managed Mutual Fund

Herman Forslund and Marcus Johnson

***Abstract*—This paper aims to reverse engineer the trading strategy of an actively managed mutual fund by identifying technical patterns in their trading. Investment strategies for many institutional investors consists of both fundamental and technical analysis. The purpose of the paper is to explore to which extent the latter can be used to predict the trading actions by taking some commonly used technical indicators as input in various machine learning algorithms to assess patterns between them and the trading of the fund. Furthermore, the technical indicators' ability to predict future prices is analysed using the same methods. The results are not sufficiently clear to suggest that the fund uses technical indicators to begin with, let alone which ones. As for the prediction of future prices, the technical indicators appear to have some predictive ability.**

***Sammanfattning*—Syftet med denna rapport är att prediktera handeln i en aktivt förvaltad aktiefond med hjälp av fyra maskininlärningsmetoder. Investeringsstrategier kombinerar i regel två analysmetoder, fundamental respektive teknisk analys. Avsikten med rapporten är att utforska huruvida det sistnämnda kan användas för att förutspå fondens handel genom att använda ett antal vanligt förekommande tekniska indikatorer och medelst maskininlärningsmetoder söka efter mönster mellan dessa och handeln. Vidare innefattar även studien en analys över hur väl tekniska indikatorer predikterar upp- respektive nedgångar på aktiepriser. Vad gäller investeringsstrategierna återfanns inga tydliga samband mellan de utvalda indikatorerna och transaktionerna. Resultaten för andra delen av studien tyder på viss prediktiv förmåga för tekniska indikatorer på marknadsrörelser.**

***Index Terms*—Machine Learning, Random Forest, XGBoost, Long Short-Term Memory, AdaBoost, Allocation Strategies.**

Supervisor: Rebecka Winqvist

TRITA number: TRITA-EECS-EX-2021:150

I. Introduction

"Who is Markowitz?" asked the fund manager. The quest to reverse engineer models that predicted the allocation of the fund's resources was off to a rocky start. Harry Markowitz is a Nobel Memorial Prize of Economic Sciences (NMPES) laureate thanks to his pioneering work in the field of portfolio theory. The fundamental axiom of asset allocation is that it is impossible to receive returns greater than a so called risk free rate without also taking on risk. The idea is that such an imbalance would lead to arbitrage opportunity, which cannot be allowed in any sensible model. Consequently, an increase of the expected return ought to be associated with greater risk.

Markowitz formalised the relation between risk and expected returns [1]. The key insight in his paper was that an investor ought to consider not only how an asset is expected to behave, but also how it is expected to behave in relation to the other assets in the portfolio [2] which is to be used to balance the assets in order to tweak the risk and expected return. He was greatly celebrated for his work, not least through the NMPES. However, the theories put forth by Markowitz don't work in practice [3]–[8]. The three key points of criticism are that the output portfolio is very sensitive to slight changes in the input, that there are a large amount of parameters that need to be arbitrarily decided (such as time periods, how often rebalancing can occur and so on) as well as that Markowitz uses variance as his definition of risk. All in all, the applicability and practical usability of Markowitz's mean variance theories is limited. The aforementioned fund manager is not alone with his attitude.

If we for one second assume that Markowitz's theories hold, determining the relative allocation of capital among the assets is a rather straightforward process using convex optimisation. However, previous attempts to reverse engineer an actively managed portfolio using convex optimisation [9], and predict a funds assessment on current markets state, yielded no fruitful results. The various practical difficulties associated with Markowitz's theories has recently called the employment of convex optimisation in financial contexts into question [10]. Besides producing vastly different allocations for small changes in predicted returns, the stability of the results is also greatly affected by the condition number of the covariance matrix. The issue is sometimes referred to as Markowitz's curse that states that more correlated assets require larger diversification which in turn generates more unstable solutions [10]. The large discrepancy between theory and practise suggests that a more empirical and less rigid approach could be better suited for the reverse engineering problem.

At the same time, the rapid progression of machine learning within finance have given rise to alternative allocation strategies [11], [12]. A shared trait among the alternative allocation strategies is the increased emphasis on momentum trading and technical indicators, often referred to as technical analysis (TA). In contrast to fundamental analysis, where the current stock price is supposed to represent the discounted value of future profits, TA solely accounts for assets price movements and trading volumes. The relevance of TA will be analysed later in the paper. Traditional institutional investors often opt for a combination of the two schools of thought [13]. However, due to the confidentiality regarding investment strategies, there is a large uncertainty regarding to what extent respective method is used in transactions, as well as which indicators are used. The reason as to why investors are secretive about their strategies is that public knowledge about them makes their trading vulnerable for exploitation, hence the interest in reverse engineering their strategies to begin with.

Problem Formulation

This study will aim to reverse engineer the trading strategy of an actively managed mutual fund using four machine learning methods including Random Forest, AdaBoost, XG-Boost and a LSTM-network, both in order to predict the trading, but also to determine which algorithm is best suited for the task. A range of technical indicators and the current portfolio allocation will be used as input to predict the trading actions of the fund. Furthermore, the predictive accuracy of employed technical indicators will also be analysed using the aforementioned machine learning methods.

II. Background

Technical Indicators and the Reverse Problem

In interviews, 87% of fund managers admit that they "put [at least] some importance in TA" [14]. The vast majority of fund mangers also reveal that TA is mainly used as a complement to fundamental analysis. Ideally, one would reverse engineer both the fundamental and technical strategy of a fund. The latter is however significantly more suitable for pattern recognition since there is a consensus regarding the definitions of the technical indicators. This study will therefore solely focus on the technical strategy aspects. The assumption is that the fundamental analysis aspect mainly determines what assets goes in the portfolio, and not the weekly trading in that asset. This is reasonable since the study also suggests that TA is the predominant factor behind trades with a weekly time horizon [14], which coincides with the trading frequency of some of the assets in the fund.

Machine Learning Algorithms

Four machine learning methods were selected for this study, a Random Forest, AdaBoost, XGBoost and a Long Short-Term Memory (LSTM) network. The three former are Ensemble Learning (EL) classifiers and the latter a deep learning algorithm. In general, tree-structured classifiers have significantly more transparency behind the decision process which is essential when the weights of individual inputs are of interest. We opted to use EL decision trees since previous studies [15], [16] suggests that they have superior predictive performance among the frequently used classifiers.

EL methods use a large number of weak learners, in this case smaller decision trees, to make predictions, which in turn has shown to improve accuracy and prevent over-fitting. Additionally, an LSTM-network was included in the comparison since several studies has suggested it is the most accurate predictor of stock-market movements [15], [17]. Whether or not this is equivalent with being suitable for predicting the behaviour of an actively managed fund is yet to be known.

As for coming Sections II. A-D, the reader should note that the primary scope of this paper is not to gain a thorough understanding of the machine learning methods below, but rather they are to be seen as practical tools for systematising the investigation.

A. Random Forest

The Random Forest (RF) algorithm, first published by Breiman in 2001 [18], uses a large set of randomly generated decision trees $\{h_i(\mathbf{x}, \Theta_i), i = 1, ..\}$, making up a random forest, to make classifications. The i:th classifier in the forest is constructed from the input $\mathbf{x}$ and the random vector Θ_i. The random vector Θ_i is a formal notation for the subset of random input features used for the ith tree. The set of $\{\Theta_i), i = 1, ..\}$ are i.i.d. random vectors. A decision tree structures decision making into a flow of questions and possible consequences, as illustrated in Fig. 1. Individual trees typically only take into consideration a subset of the available input features when node-splitting. Regarding data selection, the random forest algorithm implements bootstrap aggregating, i.e. bagging, to generate random subsets of training data. Each tree $h_i(\mathbf{x}, \Theta_i)$ is trained on a unique bootstrap training set denoted T_i. After reaching a sufficient number of trees in the forest, a majority voting decides the final outcome.

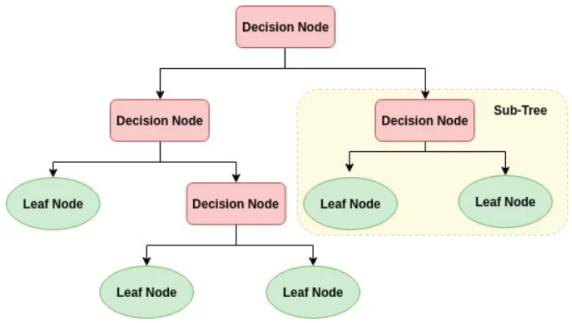


Fig. 1. A schematic illustration of a single decision tree [19]

B. AdaBoost

AdaBoost, short for Adaptive Boosting, is a gradient boosting algorithm that was first introduced by Freund and Schapire in 1995 [20]. The algorithm combines weak learners, learners that only use one input feature to make a binary classification, through an iterative process. Weights for each feature is continuously updated based on the prediction error from previous iterations. In 1999, Freund and Schapire published a mathematical formulation of the algorithm [21] which is stated below:

Given a set of samples $\{(x_i, y_i), i = 1, .., m\}$ where $Y = \{-1, 1\}$, assign each sample with a starting weight of $D_i = 1/m$. There are $t = 1, .., T$ iterations. The total error, ϵ_r, from the first iteration of weak learners, $h(x_i)$, is given by

$$\epsilon_r = P_{r \sim D_t}[h(x_i) \neq y(x_i)] \tag{1}$$

Simply put, the error is the number of miss-classifications divided by the number of points in the training set since the initial weights are uniformly distributed. However, for the following iteration steps, miss-classified samples should be multiplied with its current sample weight. Furthermore, the performance at iteration t is defined as

$$\alpha_t = \frac{1}{2} ln \left(\frac{1 - \epsilon_r}{\epsilon_r} \right) \tag{2}$$

The factor α_t is used to update the weights of the samples.

$$D_{t+1}(i) = D_t(i)e^{\mp\alpha_t y_i h(x_i)} \tag{3}$$

Note that the sign in front of the alpha varies depending on the outcome of the prediction. When, $h(x_i) = y(x_i)$, a negative sign should be used, and vice versa. This prevents the power term to become negative for miss-predictions. Finally, the new distribution D_{t+1} is normalised before the next iteration. The final prediction is, similar to that of a random forest, a majority vote of all weak learners at iteration step T.

C. XGBoost

Akin to AdaBoost and RF, eXtreme Gradient Boost (XGB, XGBoost) is an ensemble based machine learning method. The mathematical formulation is similar to, though more intricate than, that of AdaBoost [22]. Either way, it is a scalable tree boosting algorithm [22], meaning that it functions well with big data sets and works with tremendous speed. A key characteristic that separates XGB from other boosted tree algorithms is that it uses so called regularised boosting to prevent overfitting. Furthermore, it uses cross validation at every iteration of the algorithm, which allows it to stop early when it notices that more training data has sufficiently small impact on the output predictions. While computation time is not a limiting factor for the uses in this study, but rather accuracy, XGB has been included because of its success in classifying binary data [23]–[25]. XGB is widely considered to be one of the leading ML algorithms. [26].

D. Long Short-Term Memory

In its most simple form, a feed-forward neural network generates an output by traversing an input through a set of consecutive layers of computational neurons.

In contrast, a Recurrent Neural Network (RNN) contains feedback loops between its layers, which allows the network to store and reuse previous computations. This is particularly useful when the output is assumed to depend not only on the current input but on previous inputs as well.

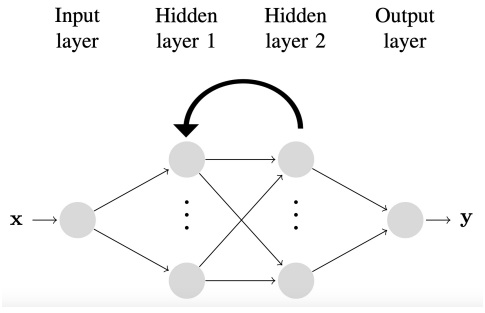


Fig. 2. A schematic illustration of an RNN [27]

In turn, a Long Short-Term Memory (LSTM) network is a specific kind of RNN. The defining characteristic of an LSTM network is that it also contains a so called *memory cell* as well as a *forget gate*. These features improves information processing which in turn enables LSTM-networks to solve problems with a higher efficiency with respect to computation time and amount of input required [28], [29]. In order to reduce overfitting, a technique called *dropout* can be used [30]. RNNs in general, and LSTMs in particular, are widely used for many different deep learning tasks, with stock price prediction being one of them [31]–[35]. Below in Fig. 3 we have a schematic illustration of an LSTM-cell

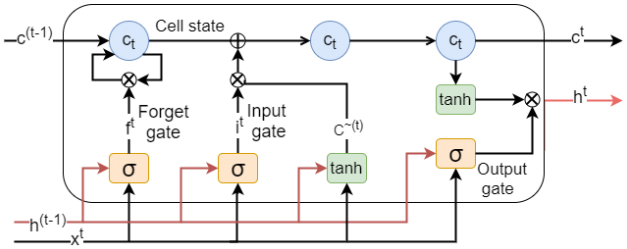


Fig. 3. A scehmatic illustration of an LSTM-cell [36]

Mathematically the model is described by the following set of equations, where we firstly introduce the net activation equation [29]

$$net_t^v = \sigma(A^v x_t + B^v h_{t-1} + b^v) \tag{4}$$

The variable A^y is the weight matrix corresponding to the input of a given gate, B^y is the other weight matrix corresponding to the output vector of the given gate, and b^y is the bias vector parameters of the given gate. Note that the weight matrices, A^y and B^y, are not explicitly marked in Fig. 3. The activation function that puts out a number on the interval [0, 1] has been chosen as the sigmoid function

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{5}$$

Equation 4 refers to the three σ-cells in Fig. 3. Hence, the equation is solved three times each iteration, once for the input gate's net activation vector (net^i), once for the output gate's net activation vector (net^y), and once for the forget gate's net activation vector (net^f). The upper index indicates which gate is referred to, and the lower index indicates the time step. We furthermore calculate the cell state vector

$$c_t = net_t^f \otimes c_{t-1} + net_t^i \otimes d_t \tag{6}$$

where $\otimes$ denotes the element wise product. The cell input activation vector, d_t, is given by

$$d_t = tanh(A^c x_t + B^c h_{t-1} + b^c) \tag{7}$$

where the superscripted c indicates a reference to the parameters of the memory cell. This activation function is defined as

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{8}$$

and thus puts out a number on the interval [-1, 1]. Finally, the output vector h_t is given by

$$h_t = y_t \otimes tanh(c_t) \tag{9}$$

The initial values of h_0 and c_0 are both equal to 0. As will become evident in section III, equations (4)-(9) are solved for every time step, i.e., for each day.

Confusion Matrix

When predicting a binary output, one way to illustrate results is with a confusion matrix. A 2x2 confusion matrix is structured as follows

$$\begin{pmatrix} TN & FP \\ FN & TP \end{pmatrix} \tag{10}$$

In the matrix, T stands for TRUE, F stands for FALSE, P stands for POSITIVE and N stands for NEGATIVE. Consequently, top left and bottom right are correct predictions whereas top right and bottom left are incorrect predictions. Since the system only changes the prediction, the resulting confusion matrix can for the most part only be shifted in left or right direction by tweaking the hyper-parameters. Another way to present the data is with the confusion vector, which which makes for easier comparisons between the matrices, and is simply defined as the $\%$ of correct guesses

$$\begin{pmatrix} \frac{TN}{TN+FP} \\ \frac{TP}{TP+FN} \end{pmatrix} \tag{11}$$

Hyper-Parameter Optimisation

The tree-based structures, in particular the boosting algorithms XGBoost and AdaBoost, takes several input parameters specifying characteristics such as the learning rate, cost function, and tree structures. Likewise, The LSTM-network requires the user to specify the number of nodes, layers and cost function. Selecting a proper set of parameters greatly influences the characteristics of the algorithm [37]. Since the parameters interact, we need to find the optimal set rather than optimising one at a time. The function GridsearchCV, which is a part of Scikit-Learn [38], has been deemed sufficiently accurate for this optimisation since further optimisation beyond this point, in the grander scheme, only yield marginally better results.

Said parameters impact the predictions, and what kind of predictions are required depends on the problem at hand. For communication purposes, the hyper-parameters are chosen so that the final confusion matrix is relatively balanced.

Definitions of Technical Momentum Indicators

In recent times, so called momentum trading has shown promising results as a trading strategy [39]. The idea is that valuations have more to do with the potential of a company, and consequently less to do with the current state of affairs. In several of the definitions below, slight modifications have been made from the original definitions, because the originals are in many cases notationally ambiguous. Moreover, the indicators output numbers on different intervals. These intervals ought to be changed so that they generate numbers on the interval $[0, 1]$. This has either been done by removing unnecessary factors in the original definitions, or by normalising the values using the normalisation formula [40]

$$y = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{12}$$

E. Interpretation of Technical Indicators

Some of the machine learning methods outlined in Section II require input as numbers, whereas other methods require one-hot encoding, meaning that the numbers ought to be converted to a binary value. All the definitions below, apart from $H.$ and $I.$, have a corresponding action signal, i.e. a conversion of the number to a buy, sell or hold signal. Prediction of patterns using the machine learning methods from Section II can be done both on the float values and on the binary signals. We use whatever yields the best results for a given algorithm.

F. Moving Average

The Moving Average (MA) on the day t with the time period length T is defined by [40]

$$MA_t^T = \frac{1}{T}\sum_{i=0}^{T} cp_i \tag{13}$$

where cp_i is the closing price of day i.

G. Weighted Moving Average

Similar to the simple MA, the Weighted Moving Average (WMA) over the time period T is defined as

$$WMA_T = \frac{2}{T \cdot (T+1)}\sum_{i=0}^{T} cp_i \cdot (T-i) \tag{14}$$

H. Exponential Moving Average

The Exponential Moving Average (EMA) is an indicator that puts larger emphasis on recent stock movements and is defined by

$$EMA_t = (cp_t - EMA_{t-1})\alpha + EMA_{t-1} \tag{15}$$

Here, α is the smoothing factor which, as standard, is chosen as

$$\alpha = \frac{2}{N+1} \tag{16}$$

where N is the number of days considered in the average. The first EMA_1 is based on the simple MA of the N previous days. When the EMA moves above the stock-price we interpret it as a buy signal.

I. Relative Strength Index

Another commonly used momentum indicator is the Relative Strength Index (RSI). It is defined as follows [41]

$$RSI = 1 - \frac{1}{1+RS} \tag{17}$$

where RS is the relative strength defined by

$$RS = \frac{EMA_{\text{updays}}}{EMA_{\text{downdays}}} \tag{18}$$

with the choice of $\alpha = \frac{1}{N}$ rather than (16), and the subscript indicates that it is the EMA of the set of up or down days, among the last 14 days.

J. Larry Williams R%

Larry Williams R% at the time point i is calculated as follows [41]

$$R\%_i = \frac{\max(p^T) - p_i}{\max(p^T) - \min(p^T)} \tag{19}$$

where p is the price, and the superscript indicates the set of all prices over the time period T.

K. K% and D%

Another indicator similar to the R% is the K% defined as

$$K\%_i = \frac{p_i - \min(p^T)}{\max(p^T) - \min(p^T)} \tag{20}$$

We also have the $D\%$ defined as the average over the last three days of the $K\%$

$$D\%_i = \frac{K\%_i + K\%_{i-1} + K\%_{i-2}}{3} \tag{21}$$

L. Volatility

The volatility of a stock is a measurement of the fluctuation in price of the stock. As mentioned in Section I, the volatility can be used as a measurement of the risk of an asset [1]. There are several methods to analyse the volatility of a stock. In this study, we use the standard deviation which is defined by

$$\sigma_t = \sqrt{\frac{\sum_i (x_i - \mu)^2}{N}} \tag{22}$$

Where x_i is the return in percent while μ is the average return in percent over the time-period N. In this instance, N is set to $N = 50$. As a general rule of thumb, investors tend to avoid volatile stocks, hence a high volatility should produce a sell signal. We classify the volatility as either a sell/buy signal based on the arbitrary limit of $\sigma = 0.06$.

M. Historic Returns

Historic returns are no guarantee of future returns. Large increases over a short time-period could however potentially suggest that the the company is overvalued while, conversely, large losses could suggest that the company is undervalued. We set an arbitrary limit of 2% per annum to classify a binary buy/sell signal, where larger than 2% returns would produce a sell-signal.

III. Method

Data Selection

The data set used for this study was obtained from an actively managed mutual fund. Due to confidentiality reasons, all identifiers to the fund will be undisclosed. The study is based on historic transaction data from a ten year period. The data in this report has been modified so that transaction frequency, transaction sizes, names of the traded assets and other unique traits are impossible to recover. Daily price data for individual stocks were retrieved from Nasdaq [42].

In order to make the analysis possible, the input data had to be modified. Cases of missing data and special instances, which corresponded to less than 15% of total transaction data, were excluded. The selection criteria were the following:

1) Only stocks were considered, excluding all other financial derivatives and thus ruling out the possibility of shorting positions. This is reasonable since the fund first and foremost invests in stocks and had few short-positions during the time-period.
2) Only public companies where data was attainable for the entirety of the holding period. This criteria excludes companies that either underwent mergers and/or became privatised during the time-period.
3) If the fund had private holdings prior to an IPO, where a public valuation was unattainable, the company was excluded.
4) All companies that were traded in other currencies than SEK were excluded. This criteria only affected a small fraction of the holdings.

Train-Test Split

In general, when using machine learning to identify patterns in data, the data is split into training and testing data. In this paper, the training was run on 80% of the data, and testing was run on the remaining 20%.

Prediction of Transactions

The prediction of transaction data was split in two segments, prediction of buying assets and prediction of selling assets. In each segment, output data was binary, i.e. either buy/do not buy or sell/do not sell for respective segment. Research suggests that TA could yield greater return when it is used for either buy timing or sell timing respectively rather than for both [43]. This, in combination with the fact that binary classification usually have a higher predictive accuracy compared to multi-class classifiers, is the main reason for the segmentation.

Accuracy of Technical Indicators

The frequency of transactions suggest that analysing the short-term predictive performance is the most relevant for this study. Accordingly, we analysed the probability at which the machine learning algorithms correctly predicted the stock price movements on a 1-day, a 3-day and a 5-day horizon. Here, a binary classification was used to categorise the stock movements as either up days, if the closing price was higher than the reference date, or down days, in the converse scenario.

Parameter Selection for Respective Cost-Function

An imbalanced data set naturally causes the algorithm to predict the most common outcome, unless there are obvious patterns in the data. To prevent this, the cost function for the minority group is amplified by a factor λ, where

$$\lambda = \frac{\#\text{total samples}}{\#\text{positive outcomes}} \tag{23}$$

This is the standard parameter choice for imbalanced sets. The impact of this choice will be discussed further in Section V.

IV. Results

Prediction of Transactions

For the different machine learning methods, predicting BUY and SELL actions for the fund yielded the confusion vectors presented in Table I.

TABLE I
CONFUSION VECTORS FOR FUND ACTION PREDICTIONS

	BUY	SELL
AB	$\begin{pmatrix}0.573\\0.492\end{pmatrix}$	$\begin{pmatrix}0.523\\0.567\end{pmatrix}$
XGB	$\begin{pmatrix}0.567\\0.505\end{pmatrix}$	$\begin{pmatrix}0.512\\0.561\end{pmatrix}$
RF	$\begin{pmatrix}0.647\\0.500\end{pmatrix}$	$\begin{pmatrix}0.505\\0.588\end{pmatrix}$
LSTM	$\begin{pmatrix}0.549\\0.602\end{pmatrix}$	$\begin{pmatrix}0.527\\0.530\end{pmatrix}$

Price Prediction Accuracy of Technical Indicators

The technical indicators used for predicting the asset price change for the coming days yielded the confusion vectors presented in Table II.

TABLE II
CONFUSION VECTORS FOR PRICE PREDICTIONS

	1d	3d	5d
AB	$\begin{pmatrix}0.515\\0.503\end{pmatrix}$	$\begin{pmatrix}0.519\\0.526\end{pmatrix}$	$\begin{pmatrix}0.517\\0.532\end{pmatrix}$
XGB	$\begin{pmatrix}0.530\\0.495\end{pmatrix}$	$\begin{pmatrix}0.537\\0.501\end{pmatrix}$	$\begin{pmatrix}0.585\\0.473\end{pmatrix}$
RF	$\begin{pmatrix}0.514\\0.515\end{pmatrix}$	$\begin{pmatrix}0.518\\0.526\end{pmatrix}$	$\begin{pmatrix}0.466\\0.589\end{pmatrix}$
LSTM	$\begin{pmatrix}0.527\\0.543\end{pmatrix}$	$\begin{pmatrix}0.545\\0.545\end{pmatrix}$	$\begin{pmatrix}0.559\\0.561\end{pmatrix}$

Feature Importance

Fig. 4 and Fig. 5 demonstrate the feature importance, i.e. the relative impact a specific feature had on the final prediction.

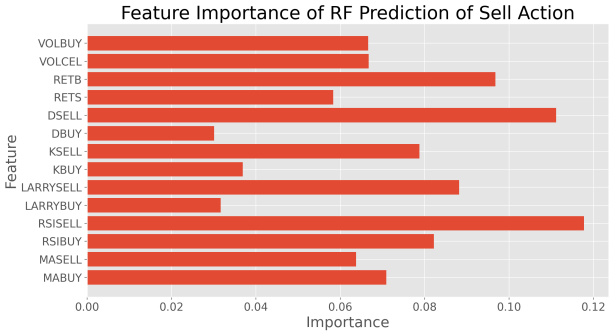


Fig. 4. Feature importance of sell prediction for the RF algorithm.

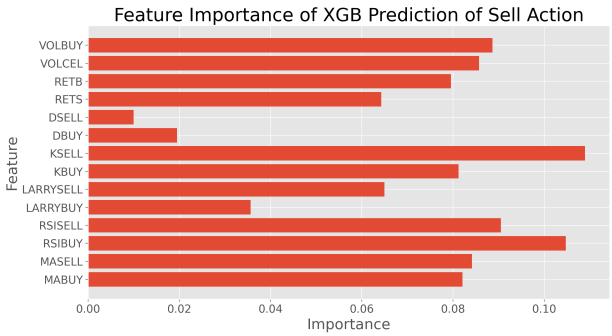


Fig. 5. Feature importance of sell prediction for the XGB algorithm.

V. Discussion and Conclusion

Impact of Cost Function and General Result Analysis

Throughout the study, the cost function was chosen as the inverse of the ratio between the number of positive outcomes divided by the total number of samples as in Eq. (21) As seen in Section IV, the performance of the algorithms was only slightly better than a pure fifty-fifty guess. The number of action predictions is vastly larger than the number of actual transactions. That indicates that a purely technical model is insufficient for predicting the trading pattern. An alternative to the used cost function would be to weigh samples equally. Since the data set was unbalanced and the correlation was low, this approach would never yield any ”action” predictions and thus limit the comparability between the algorithms.

Trading Predictions for the Algorithms

With a balanced prediction accuracy in ranges of 50-65% for all algorithms, there is no clearly superior or inferior machine learning algorithm for the task. A qualitative behaviour that is difficult to illustrate without listing numerous confusion matrices is the balance between the components of the confusion vector. The components are relatively uniform, in the sense that an increase in one component yields a decrease of similar size with regards to percentage points for the other component. With this in mind, it appears as though RF and LSTM are the best at predicting BUY actions, whereas AB and RF have the edge when it comes to predicting SELL actions. As for choosing the best algorithm for the task, the authors also note that XGB and RF are the easiest to implement. All in all, RF is considered the best algorithm for the task.

Regarding the feature importance, Fig. 4 and Fig. 5 suggest that there is no particularly important technical indicators. If the fund would base their trading on TA, we would expect the SELL and BUY signals for a given indicator to be far apart, similar to the case for the respective BUY and SELL signals of D%, i.e. DSELL and DBUY, in Fig. 4. However, there is no clear overlap of important indicators when comparing the algorithms, or in other words, different technical indicators have vastly different impact on the decision making for the different algorithms. Subsequently, no clear conclusions can be drawn. Overall, the well established indicators seemed to have a better performance compared to the arbitrarily chosen

indicators, i.e. volatility and historic returns, regardless of the chosen limits.

It is possible that the accuracy could be increased if the data was treated differently. Since companies go in and out of the portfolio somewhat frequently, the vast amount of technical data is redundant in the sense that the fund might not even monitor the stock because of a lack of interest in the fundamentals of the company. Therefore, only looking at periods surrounding the times where the fund held an asset could yield more better results.

Predictive Accuracy of Technical Indicators

The algorithms that performed the best in terms of predicting stock price increases was the RF and LSTM, which consistently had a predictive accuracy north of 50%. Although the improvements are slight compared to a fifty-fifty guess, it might be enough to outperform the index. Over long periods of time, the general market trend is almost always positive. This suggests that a random guess has positive expected value, though it is not a source of alpha. Answering the question on whether TA works or not is however more difficult. On the one hand, as N.N. Taleb argues in [8], the market valuations are driven by few unexpected outlier events that have significant impact [8]. These outlier events cannot possibly be predicted by looking at historical valuation data. TA would not be able to predict price changes whatsoever if for example a telecom company were to announce that they suddenly intend to change their business into a potato peeler company, although it would obviously impact the market value. On the other hand, market valuations are man-made, and are thus prone to manipulation, psychology and speculation [44]. One mechanism at play is that traders using TA makes it so that TA works. The results from this paper are considered insufficient to conclusively support either side, but one can at least conclude that it is not a reliable source of alpha in the form it has been used in this paper.

Other Reasons for Trading

All models are by definition simplifications of reality, and this study is no exception. Previously, we have mentioned the exclusion of short-selling as one simplification. A perhaps more significant simplification is that we omit, in absence of data, the deposits and withdrawals as well as the cash to investment ratio in the fund. The performance of an actively managed fund is measured by its returns, which incentives the fund to have a low cash to total capital ratio. A small cash deposit is however convenient since it facilitates daily withdrawals from the fund. It is reasonable to think that the fund in question actively adjusts their portfolio after significant deposits and withdrawals to maintain a sought cash to invested ratio. This would likely cause the fund to perform suboptimal trades from a TA perspective.

VI. Appendices

1) Appendix 1: Raw Data Confusions Matrices

Acknowledgment

The authors would like to thank our fantastic supervisor Rebecka ”Supreme Leader” Winqvist for providing big league knowledge and guidance.

References

[1] H. Markowitz, “Portfolio selection,” *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, Mar. 1952. [Online]. Available: https://www.jstor.org/stable/2975974

[2] E. J. Elton and M. J. Gruber, “Modern portfolio theory, 1950 to date,” *Journal of Banking Finance*, vol. 21, no. 11, pp. 1743–1759, 1997. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378426697000484

[3] R. Roll, “A critique of the asset pricing theory’s tests part i: On past and potential testability of the theory,” *Journal of Financial Economics*, vol. 4, no. 2, pp. 129–176, 1977. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0304405X77900095

[4] S. Yitzhaki, “Gini’s Mean difference: a superior measure of variability for non-normal distributions,” *Metron - International Journal of Statistics*, vol. 0, no. 2, pp. 285–316, 2003. [Online]. Available: https://ideas.repec.org/a/mtn/ancoec/030208.html

[5] R. K. Yew Low, R. Faff, and K. Aas, “Enhancing mean–variance portfolio selection by modeling distributional asymmetries,” *Journal of Economics and Business*, vol. 85, no. C, pp. 49–72, 2016. [Online]. Available: https://ideas.repec.org/a/eee/jebusi/v85y2016icp49-72.html

[6] D. Hubbard, *The Failure of Risk Management: Why It’s Broken and How to Fix It*. Wiley, 2020. [Online]. Available: https://books.google.se/books?id=fMbKDwAAQBAJ

[7] Andersson, Gustav. (2021, Apr.) Twitter. [Online]. Available: https://twitter.com/PGustavA/status/1381191039038451713/photo/1

[8] N. N. Taleb, *The Black Swan: The Impact of the Highly Improbable*, 1st ed. New York City, New York: Random House, 2007.

[9] F. R. Gustav Ekman, “Portfolio inversion: Finding market state probabilities from optimal portfolios,” *Kandidatexjobb i elektroteknik 2018, Kungliga Tekniska Högskolan, Stockholm*, pp. 121–128, 6 2018.

[10] M. L. de Prado, *Advances in Financial Machine Learning*, 1st ed. Hoboken, New Jersey: Wiley Publishing, pp. 221-223, 2018.

[11] Z. Jiang, R. Ji, and K. Chang, “A machine learning integrated portfolio rebalance framework with risk-aversion adjustment,” *Journal of Risk and Financial Management*, vol. 13, p. 155, Jul 2020.

[12] G. Chakravorty, A. Awasthi, B. Da Silva, and M. Singhal, “Deep learning for global tactical asset allocation,” *SSRN Electronic Journal*, 01 2018.

[13] L. Menkhoff, “The use of technical analysis by fund managers: International evidence,” *Journal of Banking Finance*, vol. 34, no. 11, pp. 2573–2586, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378426610001755

[14] ——, “The use of technical analysis by fund managers: International evidence,” Hannover, 2010. [Online]. Available: http://hdl.handle.net/10419/38748

[15] M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, E. Salwana, and S. Band, “Deep learning for stock market prediction,” *Entropy*, vol. 22, p. 840, Jul 2020.

[16] S. Basak, S. Kar, S. Saha, L. Khaidem, and S. R. Dey, “Predicting the direction of stock market prices using tree-based classifiers,” *The North American Journal of Economics and Finance*, vol. 47, pp. 552–567, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S106294081730400X

[17] M. Nabipour, P. Nayyeri, H. Jabani, S. S., and A. Mosavi, “Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis,” *IEEE Access*, vol. 8, pp. 150 199–150 212, 2020.

[18] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. [Online]. Available: http://dx.doi.org/10.1023/A%3A1010933404324

[19] Avinash Navlani, “Decision tree classification in python,” 2018, [Online; accessed (2021, Apr)]. [Online]. Available: https://www.datacamp.com/community/tutorials/decision-tree-classification-python

[20] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S002200009791504X

[21] R. E. Schapire, “A brief introduction to boosting,” in *Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2*. San Francisco, California: Morgan Kaufmann Publishers Inc., 1999, p. 1401–1406.

[22] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," *CoRR*, vol. abs/1603.02754, 2016. [Online]. Available: http://arxiv.org/abs/1603.02754

[23] Z. Chen, F. Jiang, Y. Cheng, X. Gu, W. Liu, and J. Peng, "Xgboost classifier for ddos attack detection and analysis in sdn-based cloud," in *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2018, pp. 251–256.

[24] S. Li and X. Zhang, "Research on orthopedic auxiliary classification and prediction model based on xgboost algorithm," *Neural Computing and Applications*, vol. 32, no. 7, pp. 1971–1979, 2020. [Online]. Available: https://doi.org/10.1007/s00521-019-04378-4

[25] S. Cerna, C. Guyeux, H. H. Arcolezi, R. Couturier, and G. Royer, "A comparison of lstm and xgboost for predicting firemen interventions," in *Trends and Innovations in Information Systems and Technologies*, Á. Rocha, H. Adeli, L. P. Reis, S. Costanzo, I. Orovic, and F. Moreira, Eds. Cham: Springer International Publishing, 2020, pp. 424–434.

[26] V. Morde and V. Anurang Setty. (2019, Apr.) Xgboost algorithm: Long may she reign! [Online]. Available: tinyurl.com/57ed6f9m

[27] D. Ekvall and R. Winqvist, "Machine learning for sleep scoring," *Kandidatexjobb i elektroteknik 2018, Kungliga Tekniska Högskolan, Stockholm*, pp. 322–328, 6 2018.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[29] F. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural computation*, vol. 12, pp. 2451–71, 10 2000.

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[31] D. M. Q. Nelson, A. C. M. Pereira, and R. A. de Oliveira, "Stock market's price movement prediction with lstm neural networks," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 1419–1426.

[32] K. Chen, Y. Zhou, and F. Dai, "A lstm-based method for stock returns prediction: A case study of china stock market," in *2015 IEEE International Conference on Big Data (Big Data)*, 2015, pp. 2823–2824.

[33] K. A. Althelaya, E. M. El-Alfy, and S. Mohammed, "Evaluation of bidirectional lstm for short-and long-term stock market prediction," in *2018 9th International Conference on Information and Communication Systems (ICICS)*, 2018, pp. 151–156.

[34] J. Mackenzie, J. F. Roddick, and R. Zito, "An evaluation of htm and lstm for short-term arterial traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1847–1857, 2019.

[35] Y. Luan, Y. Ji, and M. Ostendorf, "Lstm based conversation models," *ArXiv*, vol. abs/1603.09457, 2016.

[36] I. R. Jenkins, L. O. Gee, A. Knauss, H. Yin, and J. Schroeder, "Accident scenario generation with recurrent neural networks," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 3340–3345.

[37] N. Reimers and I. Gurevych, "Optimal hyperparameters for deep lstm-networks for sequence labeling tasks," *CoRR*, vol. abs/1707.06799, 2017. [Online]. Available: http://arxiv.org/abs/1707.06799

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[39] Z. Li and V. Tam, "A machine learning view on momentum and reversal trading," *Algorithms*, vol. 11, p. 170, Oct 2018.

[40] R. Dash and P. K. Dash, "A hybrid stock trading framework integrating technical analysis with machine learning techniques," *The Journal of Finance and Data Science*, vol. 2, no. 1, pp. 42–57, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405918815300179

[41] J. Wilder, *New Concepts in Technical Trading Systems*. Trend Research, 1978. [Online]. Available: https://books.google.se/books?id=WesJAQAAMAAJ

[42] Nasdaq. (2021, Feb.) Stockholm. [Online]. Available: http://www.nasdaqomxnordic.com

[43] N. Ekman, "An empirical analysis of the profitability of technical analysis across global markets," *Master Thesis, Lund University, Lund*, Apr 2017.

[44] Yahoo Finance. (2021, Apr.) Gamestop corp. (gme). [Online]. Available: https://finance.yahoo.com/quote/GME/

Inversion of Markowitz Portfolio Optimization to Evaluate Risk

Axel Persson and Ran Li

***Abstract*—This project investigates the applicability of the original version of Markowitz's mean-variance model for portfolio optimization to real-world modern actively managed portfolios. The method measures the mean-variance model's capability to accurately capture the riskiness of given portfolios, by inverting the mathematical formulation of the model. The inversion of the model is carried out both for fabricated data and real-world data and shows that in the cases of real-world data the model lacks certain accuracy for estimating risk averseness. The method has certain errors which both originate from the proposed estimation methods of input variables and invalid assumptions of investors.**

***Sammanfattning*—Projektet undersöker lämpligheten att använda den ursprungliga versionen av Markowitzs "Mean-Variance model" för portföljoptimering för moderna aktivt förvaltade portföljer. Metoden mäter modellens förmåga att tillförlitligt beräkna risken för givna portföljer genom att invertera den matematiska formuleringen av modellen. Inversionen av modellen utförs både för simulerad data och verklig data och visar att i fallet med verkliga data saknar modellen viss noggrannhet för att uppskatta riskpreferens. Metoden har vissa fel som både uppstår från de föreslagna uppskattningsmetoderna för inputvariabler och ogiltiga antaganden för investerare.**

***Index Terms*—Markowitz Portfolio Optimization, Efficient Frontier, Diversification, Asset allocation, Risk and Return, Inverse optimization, Inverse problems.**

Supervisors: Jacob Lindbäck & Cristian Rojas

TRITA number: TRITA-EECS-EX-2021:151

I. Introduction

A. Background

Any given investment portfolio has included a certain amount of intrinsic risk, this risk stems from various sources. When dealing with savings and investments one might think that taking fewer risks is trivial and obvious, however, a lower risk usually results in a lower expected return on the investment. Thus, striving for higher returns tends to need greater risk-taking. Typically, bonds issued by governments and corporations with a good credit rating are considered to have very low risk as these governments and corporations have a very low risk of going bankrupt. Comparably, investing in start-up companies or companies based in locations exposed to geopolitical uncertainty is much riskier since they have a considerable chance of becoming insolvent. The upside of choosing a high-risk asset is its return if it succeeds, this upside is known as the risk premium.

An investment portfolio consists of several assets. An important problem that investors tackle is which assets are to be invested in, and how much is to be invested in each of the chosen assets. One way of formulating this problem in mathematical terms is the mean-variance model, which was proposed by Harry Markowitz in 1952 [1].

The Markowitz model assumes that investors want a high return on their investments while minimizing the risk taken. As previously mentioned there is a trade-off between a high expected return and low variance, the challenge is to optimize the portfolio depending on the investor's preferences. The mean-variance model has laid the groundwork for modern portfolio theory and inspired many spin-offs which are currently responsible for many of the actively managed funds [2].

Considering the mean-variance model creates an optimal portfolio given any level of risk-averseness this project assumes that each of the portfolios investigated have assets that have been optimally allocated. The input values of the model are expected return, covariance matrix, and risk-averseness parameter. Since the expected return and covariance, matrices do not technically exist in the real world, the input values will be predicted using historical data of the stocks. The challenge in predicting returns and covariances is a science in itself and only an extremely simplified and bareboned version of it will be applied in this project.

B. Project Aim

The aim of this project is to measure the applicability of the original version of the modern portfolio theory (MPT) to real-world portfolios which are actively managed. The derivation of the method will be done by rewriting the Karush-Kuhn-Tucker conditions. By using a matrix with the expected returns, a covariance matrix of the assets, and a matrix consisting of the weights in the portfolios, the risk-averseness parameter will be the output alongside a calculated error margin and is to be compared with the "synthetic risk and reward indicator" (SRRI). SRRI is an EU-wide scale to show the risk of a fund where the scale goes from 1-7 [3]. Higher up on the scale indicates that there is a higher chance for greater return but there is also a greater chance to lose money. The calculation of the risk is based on the funds value over the last five year, that is, the calculation is based on the historical volatility [3].

C. Report Outline

Section I of the report introduces the background knowledge, project aim, and some basic theoretical knowledge which will be further discussed. Section II further discusses the model and introduces a mathematical formulation of the project alongside the crucial theory required to fully understand the procedure and result of the project. Section III explains the implementation of how the MPT can be tested on

fabricated and real-world data. In Section IV the results are presented and are being discussed alongside the implications of some of the assumptions made during the project. In Section V the results are being thoroughly discussed. Lastly, a final conclusion is presented in Section VI.

II. Literature Review

A. Markowitz's Portfolio Theory

When investing in a portfolio that consists of several assets, the anticipated profit or loss of the investment is called the expected return. The variance becomes a measurement of the risk in an investment. More specifically, the variance measure how far a number in a set is from its mean [4] (in this case, the mean of the return of an asset over a certain time period), and therefore this is equivalent to the riskiness of the investment. Consequently, Markowitz stated two rules in his Nobel prize-winning paper [1]. Firstly, the investors should maximize the expected return, and secondly, the expected return is desirable and variance is undesirable. This results in an *optimization problem* where the goal is to optimize a portfolio with assets by minimizing the risk for a given level of expected return.

If a portfolio has more than one asset the risk of the portfolio becomes more complex to calculate. The reason for this is that it does not only depend on the variance of the two assets but also how closely the returns of one asset track those of the other asset, thus the *covariance*. The covariance states how much the return of two assets change together [5].

Consider a market with k asset, indexed from 1 to k. Let x_i donate the ith asset and X be the return of all assets. The return X is assumed to be normally distributed [6] with the expected return vector μ and the covariance matrix Σ

$$X = [x_1, x_i, ..., x_k] \sim N(\mu, \Sigma) \tag{1}$$

The return of a portfolio can be written as the weighted sum of the assets, that is $Y = n_1x_1 + n_ix_i, .., n_kx_k$, where n_i is the weight of the ith asset x_i. Let $n = [n_1, n_i, ..., n_k]^T$, the return of the portfolio can be written as

$$Y = n^T X \tag{2}$$

A linear transformation of a multivariate normally distributed random variable is also normally distributed (see [7]),

$$Y \sim N(n^T\mu, n^T\Sigma n) \tag{3}$$

When the goal is to find an optimal investment strategy independent of the amount of capital invested in a portfolio the return is scaled by the sum of all the weights *n*. The weighted version of Y becomes

$$Y_p = \frac{Y}{n_{sum}} = \frac{n_1}{n_{sum}}x_1 + \frac{n_i}{n_{sum}}x_2 + ... + \frac{n_i}{n_{sum}}x_i = p^T X \tag{4}$$

where $p = [n_1/n_{sum} + n_i/n_{sum}, ..., n_k/n_{sum}]$. As n_{sum} is a scalar, maximizing Y is equivalent to maximizing Y_P. The return of the portfolio can be written as

$$Y_p \sim N(p^T\mu, p^T\Sigma p) \tag{5}$$

B. Markowitz Optimization Problem

Mentioned in section *A*, the mean-variance theory Markowitz introduced that became the basis of modern portfolio theory can be formulated as an optimization problem. The standard form of an optimization problem is [8]

$$\begin{aligned} \text{minimize} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq 0 \quad i = 1, ..., m, \\ & h_i(x) = 0, \quad i = 1, ..., p \end{aligned} \tag{6}$$

This describes the problem of finding the decision variables $x \in \mathbf{R}^n$ that minimize the so-called objective function f_0 of all the *x* that satisfy the inequalities constraint f_i and the equality constraint h_i. A point *x* is said to be feasible if the constraints are fulfilled. A feasible set can be written as

$$\{x \in \mathbf{R}^n | f_i(x) \leq 0, h_i(x) = 0\} \tag{7}$$

The standard form of optimization, as given in equation (6), can be re-written from the notation from equation (5) with the aim to find an optimal portfolio

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}p^T\Sigma p \\ \text{subject to} \quad & p^T\mu = T \\ & p_i \geq 0 \\ & p^T\mathbf{1} = 1 \end{aligned} \tag{8}$$

In the objective function, the goal is to minimize the variance of the portfolio. In the first equality constraint, the target return T for the expected return is defined. Since *p* is in percentage the second equality constraints need to be equal to one. The inequality constraint is a criterion that indicates that short selling is forbidden. The concept of short selling is that an investor borrows a stock from a broker to sell it, and then buys it back before returning it to the broker again.

With a portfolio with different kinds of assets, there will be an efficient portfolio that gives the highest expected return of a given risk. That is, for each level of risk there will be an efficient portfolio. The collection of all the portfolios can be graphically displayed as the *efficient frontier* (IV-A) [5]. The best portfolio to hold for a certain risk on the efficient frontier is the *optimal portfolio*.

C. Convexity

In order to derive optimality condition for optimality of a portfolio, which can subsequently be used to invert its risk profile, we need some theoretical tools from convex analysis. A set C is said to be convex if a line segment between any two points lies in C, where $C \subset \mathbf{R^n}$. With two points x, y in C, it can be written as

$$z = \theta x + (1 - \theta)y \in C, \quad 0 \leq \theta \leq 1 \tag{9}$$

where θ decides where upon the line *z* is found. A function f in $D \subset \mathbf{R^n} \to \mathbf{R}$ is convex for every x, y in D when

$$f(\theta x + (1 - \theta)y) \leq f(x) + (1 - \theta)f(y), \quad 0 \leq \theta \leq 1 \tag{10}$$

An important special case of a convex set is an affine set. If the line through any two points in a set $C \subset \mathbf{R^n}$ lies in C it is affine. If x and y is located in C it can be mathematically written as

$$z = \theta x + (1-\theta)y \in C, \quad 0 \leq \theta \leq 1 \tag{11}$$

An example of an affine set is hyperplanes. The standard form of a hyperplane is

$$\{x | a^T x = b\} \tag{12}$$

where $a \in \mathbf{R^n}$ and b $\in \mathbf{R}$. A hyperplane divides $\mathbf{R^n}$ into two halfspaces. A halfspace is a set of the form

$$\{x \in a^T x| \leq b\} \tag{13}$$

Since the equality constraints in equation (8) are in the same form as a hyperplane they are both affine sets. The inequality constraint has the same form as a halfspace so it is also convex. The objective function is convex because its Hessian is semidefinite [8], that is $\nabla^2 f_0(p) = \Sigma$ and $\Sigma \succeq 0$. Therefore, problem (8) is a *convex optimization problem*.

D. Slater Condition

A point x is called a Slater point if x satisfies the constraint from equation (6), that is, x is strictly feasible, meaning, $f_i(x) < 0$ for all i [9]. Further more, suppose that $f_i(x)$, $i = 1, ..., m$ are convex and differentiable, $h_i(x)$, $i = 1, ..., p$ are linear, and $\nabla h_i(x)$, $i = 1, ..., p$ are linearly independent, and the optimization problem has a Slater point. Then *Karush-Kuhn-Tucker (KKT) condition* (presented in II-E) are necessary to characterize an optimal solution [9]. This is called the *Slater condition*. This holds for Markowitz optimization problem (8) since it has only one equality constraint. As a consequence, that linear independence condition will always hold. Furthermore, all inequality constraints are convex, and the problem is strictly feasible (for instance, one can choose the portfolio $p_i = 1/k$ for all i, then $p_i > 0$).

E. Lagrangian multiplier and KKT-conditions

Suppose that a point x^* in dom (f) is optimal for a convex problem and that the slater condition holds, then there is a Lagrangian multiplier vector for x^* [10]. The idea of the Lagrangian is to take the constraint into account from equation (8) by adding weighted sums of the constraint function to the objective function. The Lagrangian is defined as

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1} \lambda_i f_i(x) + \sum_{i=1} \nu_i h_i(x), \tag{14}$$

λ_i and ν_i is called Lagrangian multipliers where λ_i is linked to the i^{ih} inequality constraint and ν is linked to the i^{ih} equality constraint [8]. With the lagrangian multiplier λ^{-1} [11] the problem (8) can be reformulated as

$$\begin{aligned} \text{minimize} \quad & -\mu^T p + \frac{\lambda}{2} p^T \Sigma p \\ \text{subject to} \quad & p_i \geq 0 \\ & p^T \mathbf{1} = 1 \end{aligned} \tag{15}$$

λ in equation (15) is called *the risk aversion parameter*. The risk aversion parameter weighs the trade-off of risk and return. The larger the risk λ is, the more risk will be penalized. Conversely, the smaller the value of λ, the less important the risk term will be in the objective function. This means, that the greater the risk an investor is willing to take, the less value the risk parameter should have. Due to the Slater condition, the KKT conditions are necessary. The conditions are [8]

Statitionarity:

$$\nabla f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* f_i(x^*) + \sum_{i=1}^{p} \nu_i^* h_i(x^*) = 0 \tag{16}$$

Primal feasability:

$$h_i(x^*) = 0, \quad i = 1, ..., p \tag{17}$$

$$f_i^* \leq 0, \quad i = 1, ..., m \tag{18}$$

Dual feasibility:

$$\lambda_i^* \geq 0, \quad i = 1, ..., p \tag{19}$$

Complementary slackness:

$$\lambda_i^* f_i(x^*) = 0, \quad i = 1, ..., m \tag{20}$$

The KKT conditions can be applied on this paper's optimization problem given in equation (15). Since p is a vector of scalars the gradient is given by $\nabla_p f_0 = -\mu + \lambda \Sigma p^*$. With the same argument $\nabla_p h(p^*) = \mathbf{1}$ and $\nabla_p f_i(p^*) = -\mathbf{e}_i$, where $\mathbf{e}$ is the standard basis $\mathbf{e} = [1, ..., 1_k]$. KKT condition for (15) becomes

Statitionary:

$$-\mu + \lambda_i \Sigma p^* + \lambda_i \mathbf{e}_i + \nu \mathbf{1} = 0 \tag{21}$$

Primal feasibility:

$$\mathbf{1}^T p^* = 1, \quad i = 1, ..., p \tag{22}$$

$$p^* \geq 0, \quad i = 1, ..., m \tag{23}$$

Dual feasibility:

$$\lambda_i \geq 0, \quad i = 1, ..., p \tag{24}$$

Complementary slackness:

$$\lambda_i p_i^* = 0, \quad i = 1, ..., m \tag{25}$$

The notation p^* indicates that it is the optimal solution for the optimization problem (15).

F. Inverse Methods

The goal is to derive an expression for the risk aversion parameter. One delimitation that has been made is that only the market that allows short selling is investigated. When short-selling is allowed the inequality constraint from the equation for the optimization problem (15) can be removed, resulting in

$$\begin{aligned} \text{minimize} \quad & -\mu^T p + \frac{\lambda}{2} p^T \Sigma p \\ \text{subject to} \quad & p^T \mathbf{1} = 1 \end{aligned} \tag{26}$$

The KKT condition (16) and the objective function from (26) leads to the expression

$$\mathbf{1}\nu + \lambda\Sigma p^* = \mu \tag{27}$$

which is equivalent to:

$$\underset{A}{\begin{pmatrix}\mathbf{1} & \Sigma p^*\end{pmatrix}} \underset{x}{\begin{pmatrix}\nu \\ \lambda\end{pmatrix}} = \underset{b}{\begin{pmatrix}\mu\end{pmatrix}} \tag{28}$$

If μ and Σ are assumed to be known, the formula (28) has two unknown variables, λ and ν. The risk aversion parameter can be determined by the least-squares solution of the linear matrix equation (28). The least-square solution computes a vector x that approximatively solves the equation $Ax = b$.

III. Implementation

Analysis was completed for both fabricated and real-world portfolios.

A. Fabricated Portfolio

A self-fabricated portfolio was constructed in order to test the method's reliability and ability to invert. The portfolio consisted of two assets, with both different expected returns. A covarinace matrix was also desigend. With two different values of the risk aversion parameter λ, two optimal portfolios will be made and illustrated on the graph of the efficient frontier (IV-A). According to the theory, the portfolio with the smaller risk aversion parameter λ should give a higher expected return but also have a higher risk.

B. Real-World Portfolios

Four actively managed portfolios are selected based on their SRRI-value, find at Morningstar (Morningstar is an American financial services firm providing investment research of most types of investments) [12]. The assets held in the various portfolios are presented in the annual reports and the interim reports of the asset managers. The dates of the reports published are in an interval between April 2020 - July 2020. With two years of historical data of the assets, find at *Yahoo Finance* [13], the parameters for the inverse problem (28) can be estimated. With a large number of assets, the covariance matrix is calculated with an analysis tool from *Excel*. To calculate the expected return matrix the adjusted close price two years back of the assets were divided by the current adjusted close price to get the percentage return. Assuming that the portfolios are optimal, the vector including the weights of the assets are selected as the same as their asset manager had chosen to use (found in the annual report). Since all of the portfolios mainly invest in the US stock exchanges, a mean-variance model which allows for short positions is implemented to find the risk aversion parameter.

The implemented method of approximating expected returns and covariance matrix is usually very sophisticated, however, in this thesis, a simpler version is performed. A trade-off between the length of history and amounts of assets included was needed, as some of the assets weren't publicly traded until recently. The authors decided that a two-year period was appropriate as it would include 95% of the total assets under management in the funds.

The input vector, including "Weights of Assets" is based on the annual or interim reports that the asset managers regularly post. While this report reflects the sizes of the positions of the portfolios, it doesn't include any transactions taken place during the two-year period. Furthermore, the liquidity of the portfolios are discarded as a "risk-free" asset would bring additional errors, the portions of the portfolios that consisted of either liquidity or assets which haven't been publicly traded for at least two years are proportionally spread out between the other assets based on the position sizes of those assets. For a full list of the excluded assets which haven't been publicly traded for two years, see appendix A.

IV. Results

A. Fabricated Portfolio

The self-fabricated portfolio was constructed including two assets. The two asset's expected returns are presented in Table I, and the covariance between the assets is presented in Table II.

Table I
Expected Returns

Asset 1	Asset 2
10%	25%

Table II
Covariance Matrix

	Asset 1	Asset 2
Asset 1	5%	10%
Asset 2	10%	30%

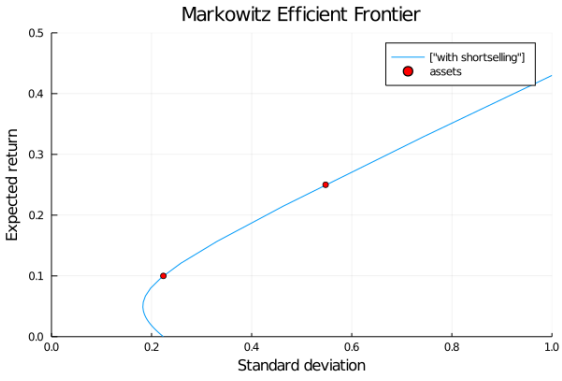


Figure 1. The efficient frontier

Table III
Results

Risk	λ
Low	3
High	0.75

The two risk aversion parameters that were tested are presented in Table III. The figure IV-A demonstrates that the

lower value of λ (which indicates less risk-averse), the further right the optimal portfolio is placed on the efficient frontier, which agrees with the theory.

B. Real-World Portfolios

Presented in Table IV, four different portfolios were selected from Morningstar with each different risk according to the SRRI-scale. The assets held in the various portfolios are presented in the annual reports and the interim reports of the asset managers. A trade-off between the length of history and amounts of assets included was needed, as some of the assets weren't publicly traded until recently. The authors decided that a two-year period was appropriate as it would include 95% of the total assets under management in the funds.

Table IV
RESULTS

Company	Fund	SRRI	λ
FundLogic	Equity Risk Manged Fund A	4	102
RBC	U.S. Equity Focus Fund A	5	164
Morgan Stanley	US Advantage Fund	6	303
Morgan Stanley	US Property Fund	7	462

As the results table clearly shows, the risk aversion parameter λ is going in the opposite direction of what it is "supposed to". Thus the data shows evidence of the mean-variance model NOT working, for today's actively managed portfolios.

V. DISCUSSION

A. Comments on Results

As shown on the results, the risk aversion parameter is reasonable for the fabricated portfolio, the lower risk portfolio had a risk aversion parameter of 3 and the higher risk portfolio had a risk aversion parameter of 0.75. However, as seen on the results table for the real-world portfolios the risk aversion parameter goes in the opposite direction as of what should be expected. The authors believe this is a random phenomenon, however, what is evident is that the mean-variance model isn't able to predict the risk aversion level of more complicated portfolios. The limitations will be discussed thoroughly below.

B. Estimation of Input Variables

The input variables of the proposed methodology are:

- Expected Return of Assets
- Covariance Matrix of Assets
- Weight of Assets

All of which are approximated with methods that are simplistic.

The expected return, μ and covariance matrix, Σ are based on daily historical prices of the assets during the past two years. The assumption that the annual expected return would be accurate by taking the two-year average return would yield high inaccuracies, especially considering a pandemic has been taking place and affecting various industries differently. The implications which the pandemic has on the financial markets is constantly being debated without any definitive conclusion. Two things are considered to be true, the pandemic increased market volatility to abnormal levels, and secondly due to the high level of capital avaliable on the market (due to travel restrictions) the markets recovered faster than "expected". The same faultiness would affect the covariance matrix similarly, as despite using daily returns as a reference, both the pandemic and the maturity of the corporation would yield significant errors, as some of the assets held are of smaller corporations which by nature are more volatile than established corporations.

The way position sizes was handled has a two-fold issue, partially it assumes the positions held at the date of the report have been constant, and partially that some assets and the liquidity are being disregarded. However, the extent this error affected the results is hard to know, the authors believe this method of implemenation should not affect the result very significantly as first and foremost the number of assets excluded is relatively small, and the portion of the liquid part of the fund was also relativey small compared to the assets included in the experiment.

C. Assumptions of the model

The Mean-variance models have a few assumptions. Some of them are palatable for reality, such as investors prefer high returns and low risk. However, some of them are less applicable in modern portfolio management [14]. For example:

1) Knowledge of the market: Markowitz assumes that expected returns, variances, and covariances of all assets are known by investors wanting to use the model. Not only is it hard to appropriately approximate accurate information of the aforementioned parameters of some assets, but to know it of all assets being traded on all stock exchanges is very misguided. Furthermore, since most portfolio managers are experts in the specific field they would choose to manage assets in these fields, eg. US Property Fund only invests in the Real Estate-industry. The implications of this is that unless portfolios are acting on identical markets it is impossible to compare them using MPT.

2) Costs: The assumption that there are no transaction costs or taxes is incorrect. Furthermore, the transaction costs might differ based on which stock exchange the assets are being traded at. Thus, even if an asset has a higher expected return with the same risk, it might not be profitable based on the cost of the transaction. Additionally, there is additional cost of taking a short position than a long position, which might be one reason why fund managers avoid them. The additional cost comes from stock loan fee which the borrowing party needs to pay.

D. Critisism of the model and its use in practice

Main criticism of the MPT being that beyond some of the assumptions that the model includes, the model still wouldn't be able to generate an optimal portfolio, as risk measurements used are probabilistic in nature and not structural.

Due to the aforementioned fallencies of the model, fund managers don't base their investment decisions on the model. Modern fund managers have very sophisticated investment

algorithms which don't solely base its decisions on expected return and covariance matrices based on historical data.

VI. Conclusion

In this project, the authors aimed to study the accuracy with which the Markowitz's Mean-variance model can predict a portfolio's intrinsic risk. Based on the original formulation of the Mean-variance model, an inversed version is derived from measuring the risk aversion parameters of given portfolios. The initial test of the model was on two simple fabricated portfolios, where the the results were logical. However, the model is proven to produce opposite results (risk aversion parameters) for the real-world portfolios. The results show that the risk aversion parameter go in the opposite direction compared to what could be assumed from theory. However, the authors believe this is a random phenomenon. What could be said for certain, is that the model inaccurately captures the risk levels of the given portfolios. One of the reasons this is the case is that the assumptions of the model has certain fallacies. An another factor is that fund managers use very sophisticated investing algorithms to make their investment decisions, thus the MPT wouldn't accurately capture the risk-aversion of the portfolios. Finally, it can be assumed that the estimated input variables had rather significant errors.

A. Future work

Improvements to the method would include more sophisticated measuring techniques for the input variables. Furthermore, since the Mean-variance model has been further modified to improve its accuracy, it would perhaps be of interest to inverse such a modification which might result in a better outcome.

Testing could be done for a model which doesn't allow for short positions.

Appendix A
Scatter plots of portfolios

Appendix B
Assets excluded in experiment

Acknowledgment

We would like to thank our supervisor Jacob Lindbäck for his help throughout the project. With his guidance and advice along with interesting discussions this project became a reality.

References

[1] H. Markowitz, "Portfolio selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, Mar 1952. [Online]. Available: http://www.jstor.org/stable/2975974

[2] S. Stoyanov, S. Rachev, B. Racheva-Iotova, and F. Fabozzi, "Fat-tailed models for risk estimation," *The Journal of Portfolio Management*, vol. 37, pp. 107 – 117, 2011.

[3] —. (2021, May) Seb. [Online]. Available: https://seb.se/privat/spara-och-placera/spara-i-fonder/risknivaer-for-fonder

[4] Hayes, Adam. (2021, Apr.) Variance. [Online]. Available: https://www.investopedia.com/terms/v/variance.asp

[5] F. J. Fabozzi and H. M. Markowitz, *The theory and practice of investment management: Asset allocation, valuation, portfolio construction, and strategies*. New Jersey: John Wiley & Sons, 2011.

[6] S. Das, H. Markowitz, J. Scheid, and M. Statman, "Portfolio optimization with mental accounts," *Journal of Financial and Quantitative Analysis*, vol. 45, no. 2, p. 311–334, Apr 2010.

[7] J. E. Gentle, "Matrix algebra," *Springer texts in statistics, Springer, New York, NY, doi*, vol. 10, p. 401, 2007.

[8] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. New York: Cambridge university press, 2004.

[9] Freund, Robert M. (2004) Optimality conditions for constrained optimization problems. [Online]. Available: Massachusetts Institute of Technology, http://ocw. mit. edu/NR/rdonlyres/Sloan-School-of-Management/15-084JSpring2004/7240EF84-B20D-419F-B1C0-2DAF3277F5C4/0/lec6_constr_opt. pdf)

[10] J. Borwein and A. S. Lewis, *Convex analysis and nonlinear optimization: theory and examples*. New York: Springer Science & Business Media, 2010.

[11] T. L. Lai, H. Xing, Z. Chen *et al.*, "Mean–variance portfolio optimization when means and covariances are unknown," *The Annals of Applied Statistics*, vol. 5, no. 2A, pp. 798–823, 2011.

[12] —. (2021, Apr.) Morningstar. Morningstar, Chicago, Illinois, USA. Moringstar. [Online]. Available: https://www.morningstar.se/se/

[13] ——. (2021, May) Yahoo finance. [Online]. Available: https://finance.yahoo.com

[14] Wigglesworth, Robin. (2018) How a volatility virus infected wall street. [Online]. Available: https://www.ft.com/content/be68aac6-3d13-11e8-b9f9-de94fa33a81e

Generation and Detection of Adversarial Attacks for Reinforcement Learning Policies

Markus Hector and Axel Drotz

***Abstract*—In this project we investigate the susceptibility of reinforcement rearning (RL) algorithms to adversarial attacks. Adversarial attacks have been proven to be very effective at reducing performance of deep learning classifiers, and recently, have also been shown to reduce performance of RL agents. The goal of this project is to evaluate adversarial attacks on agents trained using deep reinforcement learning (DRL), as well as to investigate how to detect these types of attacks. We first use DRL to solve two environments from OpenAI's gym module, namely Cartpole and Lunarlander, by using DQN and DDPG (DRL techniques). We then evaluate the performance of attacks and finally we also train neural networks to detect attacks. The attacks was successful at reducing performance in the LunarLander environment and CartPole environment. The attack detector was very successful at detecting attacks on the CartPole environment, but performed not quiet as well on LunarLander.**

We hypothesize that continuous action space environments may pose a greater difficulty for attack detectors to identify potential adversarial attacks.

***Sammanfattning:* I detta projekt undersöker vi känsligheten hos förstärknings lärda (RL) algotritmer för attacker mot förstärknings lärda agenter. Attacker mot förstärknings lärda agenter har visat sig vara mycket effektiva för att minska prestandan hos djupt försärknings lärda klassifierare och har nyligen visat sig också minska prestandan hos förstärknings lärda agenter. Målet med detta projekt är att utvärdera attacker mot djupt förstärknings lärda agenter och försöka utföra och upptäcka attacker. Vi använder först RL för att lösa två miljöer från OpenAIs gym module CartPole-v0 och ContiniousLunarLander-v0 med DQN och DDPG. Vi utvärderar sedan utförandet av attacker och avslutar slutligen med ett möjligt sätt att upptäcka attacker. Attackerna var mycket framgångsrika i att minska prestandan i både CartPole-miljön och LunarLander-miljön. Attackdetektorn var mycket framgångsrik med att upptäcka attacker i CartPole-miljön men presterade inte lika bra i LunarLander-miljön.**

Vi hypotiserar att miljöer med kontinuerliga handlingsrum kan innebära en större svårighet för en attack identifierare att upptäcka attacker mot djupt förstärknings lärda agenter.

***Index Terms*—Deep Reinforcement Learning, Adversarial Attacks, Adversarial Attack Detection, Fast Gradient Sign Method, Deep Deterministic Policy Gradient, Deep Q-Learning, Likelihood Ratio Test, CUSUM**

Supervisors: Alessio Russo, Division of Decision and Control Systems, EECS School.

TRITA number: TRITA-EECS-EX-2021:152

I. Introduction

Machine Learning (ML) has become a very fast improving area of research due to its great capability to learn from data without being explicitly programmed. Reinforcement learning is an area of machine learning in which the goal is to decide the actions taken by an agent, working in a so-called environment, so as to maximize the total collected reward. Reinforcement learning is a widely used method for companies to solve different problems. For example, it is used by Tesla in their self-driving cars [1].

Another example where reinforcement learning can be used is the game of chess. The states is represented by certain configuration of the pieces on the board. In each round the agents can move a piece (which is to perform an action). Each move provides a reward to the agent. For example, we might get a large reward for taking a rook with our queen. The goal of reinforcement learning is to learn a policy to maximize this reward, meaning a strategy of which actions to take in a given state. See [2], [3] where deep reinforcement learning techniques are used to create chess playing agents.

There are several ways of solving the problem of finding policies (such as Q-learning or the SARSA-algorithm, see Sutton and Barto [4]) but it becomes quite difficult when the state and or the action space takes on continuous values, meaning there are an infinite number of actions to take or states to be in. In recent years a solution to these types of problems has been proposed by using what is called deep reinforcement learning, in which neural networks are used to model for example the Q-function (DQN, see Silver et al [5]) or the policy itself (DDPG, [6]). This has led to great results since neural networks are very apt at generalizing, and are therefore able to learn the Q-function or the policy, even for values not seen during training. A thorough source for deep reinforcement learning can be found here [7]. As great as these kinds of solutions are however, it leaves the model open to so called adversarial attacks based on the fact that neural networks have been shown to perform very poorly on adversarial examples [8]. An adversarial example is input data that has been tampered with in a certain way (eg. by using the FGSM method, se Goodfellow et al. [8]) so that the result is often indistinguishable to the human eye, but completely fools the network. It is therefore of great interest to investigate how models trained using deep reinforcement learning are affected by these kinds of attack. Another very interesting topic is the use of detectors which can be trained to find out if the model is currently under attack. Additional information about adversarial examples can be found here [9].

II. THEORY

A. Markov Decision Process

A Markov Decision Process (MDP) is a time-discrete stochastic process that can be used to model discrete-time stochastic processes. Mathematically, an MDP is described as a tuple (S, A, P, r). The first element of the tuple, S, stands for the state space, which is defined as the set possible states of the environment. The second element is the action space, or the set of all possible actions that can be taken by the agent. Thirdly, the transition function $P(s, a, s')$ is a function $P : S \times A \times S \to [0, 1]$ that returns the probability that the agent will end up in state s' given that it starts in state s and takes action a. Lastly, there is the reward function $r(s, a)$ which returns a reward (usually a real number) given a state s and an action a.

Solving a Markov decision process means finding a so called optimal policy. A policy in general is simply a mapping $\pi : S \to A$, that is mapping from states to actions. The optimal policy is the policy $\pi^\star$ that maximizes the expected collected reward (which means not only choosing the best action considering the current state, but also considering future rewards). Since we usually aim to maximize the total collected reward from time 0 to infinity, we introduce a discount factor $\lambda \in (0, 1)$ that is used to tune how myopic the agent is. Therefore, the optimal policy should maximize the following expression 1 for any initial state s_0.

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \lambda^t r(s_t, a_t) | a_t = \pi(s_t)\right]. \tag{1}$$

B. Reinforcement Learning

In this project we make use of different deep reinforcement learning techniques ([6], [5], [7]) but for the sake of completeness, we first briefly explain what reinforcement learning is.

In reinforcement learning we don't have full knowledge of the dynamics, which means that we don't know the reward function and we don't know the transition function. The goal is to come up with ways to find the optimal policy while not knowing these functions. This often means letting the agent take lots of actions in order to explore how the system behaves. Many techniques to do this can be found in Sutton and Barto [4].

Some very important functions that are often used in reinforcement learning are the value function and the Q-function (see equations 2 and 3).

$$V^\pi = \max_a \left[r(s, a) + \lambda \sum_{s' \in S} P(s'|s, a) V^\pi(s') \right]. \tag{2}$$

The value function corresponding to a policy π is simply a function $V : S \to \mathbb{R}$ that returns the expected total discounted reward given that the agent starts in state s and follows policy π. The optimal value function $V^\star$ is sometimes also simply referred to as the value function. This refers to the value function of the optimal policy $\pi^\star$.

$$Q^\pi(s, a) = r(s, a) + \lambda \mathbb{E}_{s' \sim P(\cdot|s,a)}[V^\pi(s')]. \tag{3}$$

The Q-function is similar to the value function, but it is instead a function $Q : S \times A \to [0, 1]$ that also considers the value of an action. This means that the Q-function returns the expected total discounted reward given that the agent is in a state s, takes action a (which does not need to be in accordance with policy π) and after this always follows policy π. Also here the optimal Q-function $Q^\star$, simply refers to the Q-function corresponding to the optimal policy $\pi^\star$.

Reinforcement learning often revolves around different techniques of learning the value function, Q function or the policy itself (Sutton and Barto [4]). Here however, we use deep reinforcement learning techniques, which requires the use of deep neural networks.

C. Deep Reinforcement Learning

1) Deep Q Learning (DQN): The first deep reinforcement algorithm we look at is called deep Q-learning (DQN, see Mnih et al. [5]). In this method we use a neural network to try to model the Q-function, which is a function that assigns the value of the reward that we expect to collect given a certain state s_t and a certain action a_t. DQN assumes that the state space is continuous but that the action space is discrete, hence what we wish to achieve is a trained network that takes in a state vector, and returns a vector with the Q-values for every action in the action space (which works since they are discrete). The agent then simply picks the action with the highest Q-value depending on which state it's in.

To train the Q-network we make use of a so called replay buffer. This is a buffer that saves the experiences (meaning tuples containing the initial state s_t, the action taken in that state a_t, the reward collected r_t, and the new state the agent ended up in s_{t+1}) and lets the network sample from it, so that it can learn. This way the agent saves new experiences in the buffer, while training on the current experiences already in store. This has the the benefit of making it possible for the network to learn from an experience more than once, while also making the data seem more i.i.d which helps with convergence while training.

2) Deep Deterministic Policy Gradient (DDPG): The second deep reinforcement learning algorithm is called deep deterministic policy gradient (DDPG). This is a so called actor-critic method, in which not only the Q-function is modeled, but also the policy itself, hence two neural networks are used, called the actor and the critic respectively. DDPG assumes that that both the action space as well as the state space takes on continuous values, as opposed to DQN. It is important to note that DDPG, as well as DQN, is a deterministic algorithm, which means that given a state the same action will always be taken. This brings up problems in regards to exploration, which have to be solved by adding some random noise to the action during training. For DDPG a replay buffer is also used for the same reasons as stated for DQN. DDPG was introduced by Lillicrap et al [6], and can solve problems not possible for DQN because of the

Algorithm 1 Deep Q Learning with Experience Replay

Initialize memory R
Initialize action-value function Q with random weights θ
Initialize target action-value function Q' with random weights θ'
for episode = $1, \ldots, M$ **do**
  Reset initial state s_1
  for $t = 1, \ldots, T$ **do**
    With probability ε select a random action a_t
    Otherwise select $a_t = \text{argmax}_a \, Q(s_t, a|\theta)$
    Execute a_t in environment
    Observe reward r_t and state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in R
    Sample batch of transitions (s_j, a_j, r_j, s_{j+1})
    Set $y_j = r_j$ if episode terminates at $j + 1$
    Otherwise Set $y_j = r_j + \lambda \max_{a'} Q'(s_{j+1}, a'|\theta')$
    Perform a gradient decent step on $\sum_j (y_j - Q(s_j, a_j|\theta))^2$ with respect to network parameters θ
    Every C steps set $Q' \leftarrow Q$

continuous action spaces. Also here two neural networks are used (four in total) called the target and the main network. The target network is not updated as often as the main network which leads to much better stability of convergence during training.

Algorithm 2 Deep Deterministic Policy Gradient

Randomly initialize critic network $Q(s, a|\theta_Q)$ and actor $\pi(s|\theta_\pi)$
Initialize target networks $Q'(s, a|\theta_{Q'})$ and $\pi'(s|\theta_{\pi'})$ and set weights equal to main networks
Initialize replay buffer R
for episode = $1, \ldots, M$ **do**
  Initialize random noise N
  Reset initial state s_1
  for $t = 1, \ldots, T$ **do**
    Choose action $a_t = \pi(s_t|\theta_\pi) + N_t$
    Take action a_t, observe reward r_t and new state s_{t+1}
    Store experience (s_t, a_t, r_t, s_{t+1}) in R
    Sample a random mini batch of n transitions (s_t, a_t, r_t, s_{t+1}) from R
    Calculate $y_i = r_i + \lambda(Q'(s_{t+1}, \pi'(s_{t+1}|\theta_{\pi'})|\theta_{Q'})$
    Update Critic by minimizing: $\frac{1}{n} \sum_{(s,a,r,s')} (Q(s, a|\theta_Q) - y)^2$
    Update the actor with policy gradient: $\nabla_{\theta_\pi} = -\frac{1}{n} \sum_s Q(s, \pi(s|\theta_\pi)|\theta_Q)$
    Update the target networks:
    $\theta_{Q'} \longleftarrow \tau\theta_Q + (1 - \tau)\theta_{Q'}$
    $\theta_{\pi'} \longleftarrow \tau\theta_\pi + (1 - \tau)\theta_{\pi'}$

D. Fast Gradient Sign Method

FGSM is a method of producing adversarial examples, and it stands for Fast Gradient Sign Attack. Given a trained neural network, and some input data x, we want to find a small perturbation Δx such that $x + \Delta x$ diminishes the performance of the network as much as possible. The way to do this is by first running the data point x through the network and retrieve the output y. In the case of classification y is assumed to be some sort of probability distribution. We assume that the network is well trained, meaning that we put the target as only zeroes with a one on the class with the highest probability in y. This is then used to calculate the cross entropy loss. The gradient with respect to the loss function is then taken, which is used to modify the original input. The gradient of the loss function determines in what way we should modify the state to maximize the loss. This is added to the original output, and can be seen as a kind of noise, a slight perturbation, that maximizes the cost and reduces performance of the trained network. This can be modelled as equation 4

$$\tilde{x} = x + \varepsilon\phi\left(\nabla_x J(\theta, x, y)\right). \tag{4}$$

Here, ϕ is the sign function, meaning it returns one for inputs greater than zero, and negative one for values lesser than zero. $\tilde{x}$ is the adversarial example created by the FGSM method, x is the original input, the original state, ε is a scalar between one and zero ensuring perturbations are small. J is the loss function which has θ, the model parameters, x, the original state vector and y, the estimated correct action that should be taken by the agent as input. More on FGSM can be found here [8].

Equation 4 however is used while performing attacks on networks used for classification. It assumes that the input x consists of values between [0,1], or [-1,1]. This has to be adjusted for our purposes, since in CartPole and Lunar lander the states can both take on very large and small scalars. To do so, we introduce an alternative formulation where we scale each component of the state vector by at most ε, as follows

$$\tilde{x} = x \odot \left(1 + \varepsilon\phi\left(\nabla_x J(\theta, x, y)\right)\right). \tag{5}$$

$\odot$ means that the product is done element wise. This is done to ensure that the perturbation doesn't change the input x with more than $100\epsilon\%$.

III. METHOD

A. Solving CartPole-v0 with DQN

In CartPole-v0 a pole is attached to a cart which can move along a frictionless track. The objective is to move a Cart to balance a pole that is stuck to the cart, and the possible actions are to push the cart with a certain force either to the left or to the right. The state space is a subset of $\mathbb{R}^4$ and is continuous, whereas the action space is discrete and one-dimensional with two possible actions. Since this problem has a continuous state space, and a discrete action space, we solve this environment by using DQN.

We trained three different policies improving the Q function using DQN, until the environment was solved. To solve the environment the policy has to hold the pole upright for an average of 195 time steps over one hundred episodes, which

all three policies managed to achieve. We trained the policy for 10 000 episodes, but stopped the training if the average score over the last one hundred games was above 194 points. We tried to adjust the learning rate, buffer size and the number of hidden layers for both the target network and the action target network. The parameters we chose can be seen in table I. These parameters were used for training three different networks, changing the random seed in between every run. All policies managed to solve the environment.

TABLE I
THE CHOSEN PARAMETERS SOLVING LUNARLANDER AND CARTPOLE

Parameters for	CartPole	Lunarlander
Learning rate	0.0001	0.00003
Batch size	128	64
Number of episodes	10 000	1000
Discount factor, λ	0.75	0.99
Nodes in first hidden layer	256	300
Nodes in 2 hidden layer	256	400
τ	None	0.001

B. Solving LunarLanderContinuous with DDPG

The LunarLander environment is a little more complicated than the CartPole. The state is here represented by a 8-dimensional vector, whereas the actions are two-dimensional, and both take on continuous values.

DQN assumes that we have a continuous state space but the action space still needs to be discrete, which means that a different algorithm is needed to solve this environment. Hence, we chose to use DDPG, which is an actor-critic method where we use two different neural networks called the actor and the critic. The actor models the policy itself, meaning it takes in a state vector and returns an action vector to be taken given that state. The critic tries to model the Q-function, which is used to evaluate the actions chosen by the actor network.

We trained four different policies, which all managed to solve the environment. The criteria required to solve it can be found on OpenAI's website, which is to get an average reward of 200 over 100 consecutive episodes.

The actor network had one hidden layer with 300 hidden nodes and the ReLU function as activation, as well as batch normalization. It then had a second hidden layer with 400 nodes, batch normalization and ReLU as activation. Lastly, it had a two-dimensional output layer with the $\tanh$ activation function as activation.

The critic network on the other hand takes in 10 dimensions, 8 for the state and 2 for an action. The action part of the input is first put through a linear layer with 400 neurons and the ReLU activation function. The state part gets put through two linear layers with ReLU activation and using batch normalization twice (before feeding it through the layers). The first layer have 300 neurons and the second one 400 neurons. Both of these are then put into one vector, ReLU is applied one more time and they both gets run through a final linear layer, without activation function.

The hyper-parameters we used are displayed in table I. Training was done in batches with a batch size of 64, and we trained for 1000 episodes. Training did not always result in a solution, but after some tries we managed to get four different policies that solved the environment by changing the random seed between every training run.

C. Producing Adversarial Examples for CartPole

We used the trained policies to produce adversarial examples. To produce adversarial examples we used the adjust equation 3, as mentioned earlier. Important to note is that the Q-network used in DQN returns Q-values and not a probability distribution. To make FGSM work as described in the preliminaries we then applied the softmax function, described in equation 6, which transforms the output to a probability distribution.

$$y(s) = \frac{e^{Q_\omega(s)}}{\sum_{i=1}^{k} e^{Q_\omega(s)_i}}. \tag{6}$$

Equation 6 is the general softmax function for a classifier with k different classes, so in our case $k = 2$. The attacks were performed both on the agents used to produce them, as well as on the other agents in order to see how the attacks transfer to different solutions of the same problem. To see how well the FGSM examples affected the policies, each policy was tested for various values of ε. The values used for ε were $\{0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40\}$. To see how the attacks transferred to different policies, each network was used to generate attacks, and then these attacks were used on both of the other networks, for all the values of ε.

D. Producing Adversarial Examples for LunarLander

For the LunarLander environment the cost function of the actor network is the critic network itself. Hence, to use FGSM, the gradient with respect to the input needs to be computed, but the state is also used in calculating the target for the cost function, so the chain rule needs to be applied.

$$-\nabla_\theta Q_\omega(x, \pi_\theta(x)) = -\nabla_a Q_\omega(x, a) \nabla_\theta \pi_\theta(x)|_{a=\pi_\theta(x)} \tag{7}$$

This gradient is quite computationally heavy, but we still managed to generate the attacks. The attacks were generated for different values of ε. The values used were $\{0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40\}$.

E. Training of Binary Neural Network Classifiers

In order to detect attacks we chose to build a binary neural network classifier with two classes, one representing that an attack is occurring and the other that it's not. We wanted the output of the network to be a number between 1 and 0, representing the probability that the network gives of an attack taking place. When the output of a network is a probability distribution, a very useful loss function is,

$$J(\hat{y}, y) = -\sum_{i}^{k} y_i \ln(\hat{y_i}), \tag{8}$$

called the cross entropy loss function. Because of these reasons, we chose to train the networks using binary cross

entropy loss, described in equation 9. This is the same as the loss in equation 8 with $k = 2$.

$$J(\hat{y}, y) = -y \ln(\hat{y}) - (1 - y) \ln(1 - \hat{y}). \tag{9}$$

Here, y is the label of the data point, hence a zero or a one, and $\hat{y}$ is the output of the network, which is a number between zero and one.

Since the data here is sequential, alternating between a state vector and an action vector, we choose to make the data points into longer vectors with five sequential state action pairs put next to each other as can be seen in equation 10. This is because important information lies in the sequence of states and actions. The labels were simply a one if the data point was generated while under attack, and a zero otherwise.

$$x_i = [s_i, a_i, ..., s_{i+4}, a_{i+4}] \tag{10}$$

To do different tests we generated many different data sets. First, three data sets were generated for both environments where all of the data was drawn from a single policy network (we used three different policies for both environments). After that we generated three new data sets for both environment, where the data now was drawn from two of the three policy networks. Also important to note that the number of data points used for the LunarLander environment was around 750 000 for all six data sets, and only about 30 000 for the data sets for CartPole. This is because we assumed that the task of detecting the attacks would be much harder since LunarLander only has continuous values as well as higher dimensionality of the data.

In addition to this we also generated 3 validation sets per environments, also here using data from two data sets at a time. The reason we didn't simply divide the already gathered data sets into training and validation sets is because the data was, as already explained, gathered from a stream of data, so information would have carried over from the training to the validation sets, and we would have gotten a much higher score on our validation set than would have been true to completely new data.

For the training of the attack detection network, we chose to use a simple multilayer perceptron with a single dimensional output. The input dimension was different for the policies of the different environments, 50 dimensional for LunarLander and 25-dimensional for CartPole. The activation functions used were ReLU for the hidden layers and the sigmoid function for the output, which then makes the output be a number between one and zero. This is viewed as the probability that the policy is under attack, with the threshold for deciding weather and attack is occurring or not being 0.5.

For each environment we then trained three classifiers where the formerly mentioned training and validation data sets where used in training. The detection of attacks for the LunarLander environment required more complex network layouts and we there employed a neural network with four hidden layers using the ReLU activation in every layer as well as 256 hidden neurons. The output layer was, as previously mentioned, one-dimensional and the sigmoid activation function was used to get an output between zero and one. The training for the LunarLander detectors were done with mini batch gradient descent. The batch size used was 528 and training was continued for 100 epochs (or until the validation error started going up) with a learning rate of 0.0001. The loss function used was the already mentioned binary cross entropy loss, which is described in equation 9.

For the classifiers of the CartPole environment our networks were considerably simpler. We employed networks with two hidden layers with only eight neurons each, and ReLU as activation. The output layer was again one dimensional with the sigmoid function as activation. Training was with normal batch gradient descent (no mini batch meaning the network weights were only updated once per epoch) and was continued for 2400 epochs or until the validation error started increasing, with a learning rate of 0.0013.

F. Online Detection of Adversarial Attacks

It is interesting to see how the detectors perform on live data (meaning data that is generated continuously and not data that has been gathered during some period of time) since this is a more realistic setting. Hence, after training several binary classifiers as mentioned earlier, we tested each binary classifier during real time where we could turn on and off attacks at any time. In order to detect attacks live we used the CUSUM statistic (Basseville [10]) that can be found in equation 11.

$$G_t = \max(0, G_{t-1}) + \ln\left(\frac{\phi(x)}{1 - \phi(x)}\right) \tag{11}$$

Here, $\phi(x)$ is the output of the classifier with respect to a data point x.

Each classifier was exposed to attacks both from policies it had seen during training, and attacks drawn from policies it had not seen during training. This was done both for the Lunarlander and CartPole environments.

IV. Results

A. Adversarial Examples On CartPole-v0

Figure 1 summarizes the results of average score in the CartPole environment during adversarial examples. The blue line depicts the case where adversarial examples have been drawn from the network under attack, and the orange line depicts the case where a different policy has been used to create adversarial examples.

Figure 2 summarizes the results of standard deviation in the CartPole environment.

B. Adversarial Examples on LunarLander

In figure 3 results from how the adversarial attacks diminished the performance of the agent are shown. We show both how an agent is affected by adversarial attacks where the attacker used the agent itself to generate the attack with FGSM, as well as how an attack generated using one agent affects a different agent that has been trained to solve the same task. Both curves seem to follow each other fairly well as a function of the epsilon that was used in FGSM.

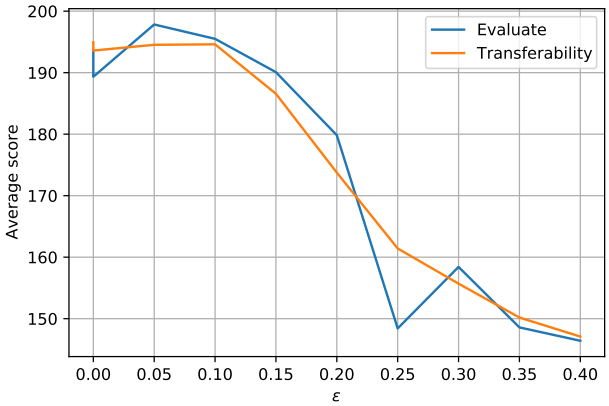


Fig. 1. The means of the average rewards of the policies solving CartPole as a function of the epsilon used during the attacks.

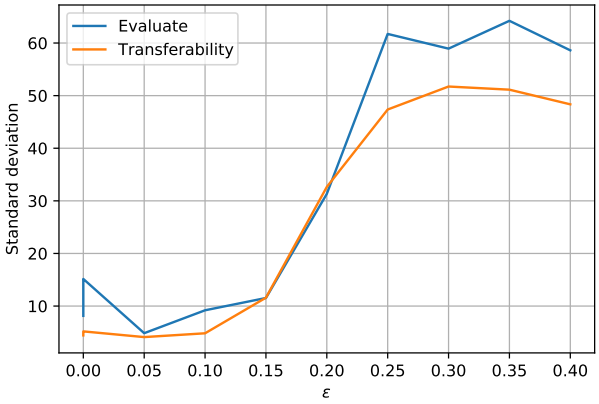


Fig. 2. The standard deviations of the average rewards of the policies solving CartPole as a function of the epsilon used during the attacks.

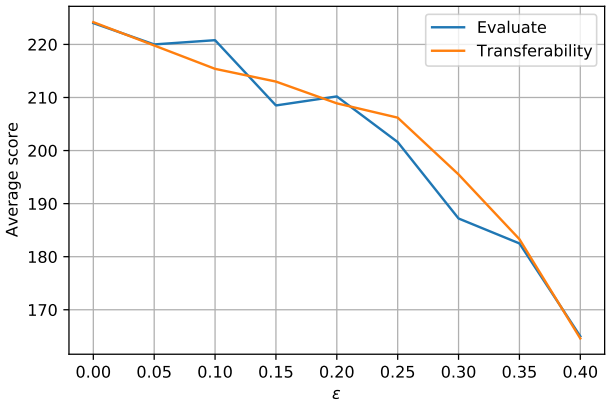


Fig. 3. The means of the average rewards of the policies solving LunarLander as a function of the epsilon used during the attacks.

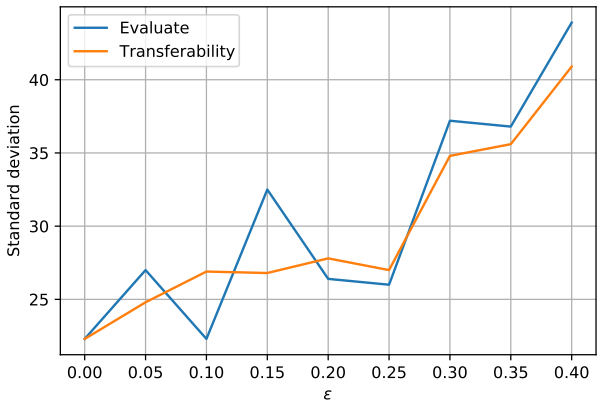


Fig. 4. The standard deviations of the average rewards of the policies solving LunarLander as a function of the epsilon used during the attacks.

The curves in figure 3 are the means from four different networks. In the transferability curve every agent was tested on attacks from all of the three other networks.

In figure 4 the standard deviations are shown, which get bigger and bigger the bigger the epsilon used in the attack.

C. Adversarial Attack Detection

1) Attack Detection for CartPole-v0: In CartPole-v0 the binary classifier for the environment achieved a mean of 98.56% correctly classified data points , and a standard deviation of 0.0037, on data from the same networks the binary classifier had been trained on (although not the same trajectories). When the binary classifier was tested on attacks performed by network never seen by the classifier, meaning it had not been trained on similar data during the training process, it achieved a mean of 97.81% correctly classified data points and a standard deviation of 0.0073.

2) Attack Detection for LunarLander: The three detectors trained to detect attacks on the policies solving the LunarLander environment had an average accuracy of 90.33% with a standard deviation of 0.47% on the test sets with data drawn from the same networks used during training (although not the same data set). The accuracy went down to a mean of 60.94% with a standard deviation of 3.59% when testing on data drawn from a network not used in training.

3) Online Detection for Both Environments: At last we tried to determine how well this model works online as discussed earlier. The results are as following. The graph describes the CUSUM statistic (see Basseville [10], also equation 11, where $\phi(x)$ is the output of the classifier). This is to make it grow very fast if the classifier detects an attack. The statistic on the y-axis is calculated according to equation 11. The value on the y-axis should be growing while being under attack. We started detection with attacks being turned off and turned them on after half of the episodes were done. For the CartPole detectors this happens roughly about the same time since the CartPole environment stops after 200 time steps and the

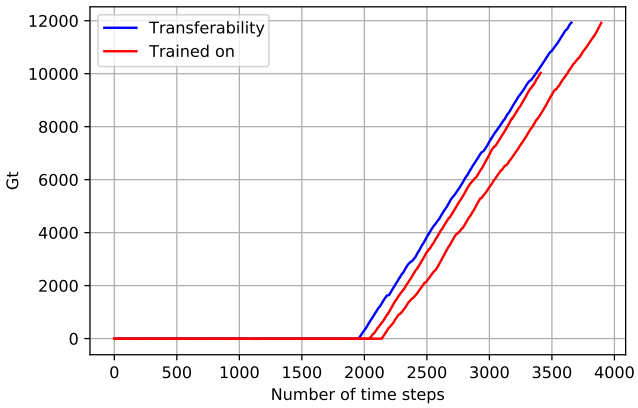


Fig. 5. The results of online detection for binary classifiers trained to detect attacks on the policies solving the CartPole environment.

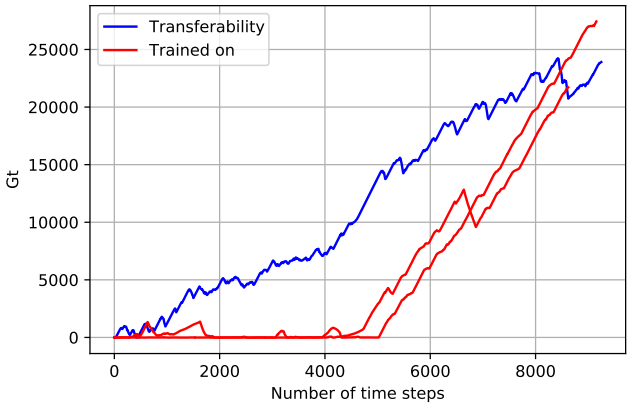


Fig. 6. The results of online detection for binary classifiers trained to detect attacks on the policies solving the LunarLander environment.

policies usually achieve a very high score, however for the LunarLander detectors it is more spread out when the attacks start since there is a very large variation in the number of steps the agents take for an episode of landing the lunar lander. The results for CartPole can be seen in figure 5 and for LunarLander in figure 6.

V. Discussion

A. Comments on the training of the models

The training of the policies did not pose very big problems. The most time consuming part was to find good hyperparameters (learning rate, network layout, discount factor etc) that allowed for easy convergence of the training. The time it took to train with DDPG also made this more difficult, but once good parameters were found, it was quite easy to train multiple policies, simply by adjusting the random seed before training. Changing the random seed before training was done in an attempt to reduce the probability of training two identical policies solving the same environment.

B. Thoughts on Results of FGSM attacks

The FGSM attacks worked as planned with the average reward going down quite drastically. We did however apply quite large values of ϵ during the attacks (see equation 4) meaning that the disturbance during the adversarial attacks were somewhat large as opposed to the ones used in Huang et al [9]. However, our state spaces different than the ones used in that paper, which may be a contributing factor to why we needed greater epsilons to get FGSM to work well.

C. Transferability

The transferability results where not exactly what we would have expected. We assumed that the average score would diminish quite drastically when applying FGSM with examples drawn using the same network, however we didn't expect the effect to be as big when applying the adversarial examples from one network to another one. Our results show however that the performance of the agents seem to go down fairly similarly when applying it to different networks solving the same environment. This seemed to be the case for both environments which is a little surprising. One possibility is that this shows that the policies are quite similar, and hence FGSM for one policy translates very well to other ones. This would be strengthened by the fact that these environments aren't overly complicated, however we would have expected to see a bigger difference at least for the LunarLander environment since it is quite more complex than CartPole and hence the policies are less likely to be very similar. For CartPole it would not be too surprising that policies that solve the environment behave very similarly since there are only two discrete actions to take. More often than not one is clearly going to be the wrong one to take, which makes the different agents behave very alike.

Another possibility as to why our results on transferability are very similar to the normal FGSM attacks are the bigger size of our epsilons. If we disturb the states with a very large disturbance it will clearly change the average reward very drastically no matter how we choose the perturbation. This together with the fact that we needed fairly big epsilons to get the performances to go down could possibly be why we get very similar results.

D. Attack Detection

As can be seen in the results, we managed to train a seemingly very good attack detector for the CartPole environment. The attack detector for CartPole is able to predict to a very high accuracy if an attack is taking place or not, even if the attack is produced with FGSM on the different network(s) that was used during. The very high accuracy on the test set, and the transferability set, together with the relatively simple action space of the environment, indicate as we alluded to in the previous section, that our policies are very similar. This would explain how well our fairly simple detector network is able to perform on all the FGSM attacks we used.

The results for the detectors on LunarLander however are considerably worse, even though we use a more complex neural network layout as well as about 20 times more data

points. The continuous nature of the action space clearly has a large impact on how easy it is to train the detector. To follow the thought processes we had about the CartPole detector at large this shows that the policies most likely are very different from each other here. This is also strengthened by the fact that we do get fairly good results on FGSM attacks taken from the same policies used during training, but barely better than random when applying it to data taken from completely different agents. On top of all this, it is very hard to analyze neural network since we don't really know how they actually solve the problem (hence the very popular saying that they are like a black box). It is therefore hard to say with certainty exactly why we don't manage to train a better classifier for the detectors on LunarLander, and it is very possible that it is doable with some better knowledge on how to train them effectively. It is worth noting that by increasing the amount of training data the results get quite a bit better, so that could possibly also be something that could be improved upon.

E. Future Work

There are many ways in which this project could be continued. The obvious one is to increase the amount of environments solved as well as the algorithms used for the different environments. It could also maybe be possible to solve every environment with every algorithm used, and hence see how well FGSM attacks transfers not only between policies but also between algorithms. In this case it would be necessary to divide the action space into bins in order for DQN to be able to solve LunarLander, but that could easily be fixed.

It is also possible to improve the training of the attack detectors. There are many advanced techniques to train neural networks that could be implemented to improve results. An interesting thing could also be to change the architecture and implement a recurrent neural network (RNN) instead of a normal feed-forward network. Recurrent neural networks are made to handle temporal data, and since that is what we are dealing with here, then that could possibly improve results.

VI. Conclusion

This has been a very interesting project and it has gone rather well. We have implemented two different deep reinforcement learning techniques (DQN and DDPG) to environments from OpenAI. We have then used these solutions to produce adversarial examples (by using the fast gradient sign method) to see how this affects the performance of our solutions, as well as how it transfers to different solutions of the same problem with the same algorithm. Lastly we looked at, and succeeded fairly well at, building binary neural network classifiers to use as attack detectors on our already trained agents.

Acknowledgment

The authors would like to thank our supervisor Alessio Russo for helping to plan and guide the project as well as for providing lots of valuable help with the implementation.

References

[1] "Autopilot kernel description," https://www.tesla.com/autopilotAI, accessed: 2021-05-12.

[2] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, "Mastering chess and shogi by self-play with a general reinforcement learning algorithm," *arxiv:1712.01815*, 2017.

[3] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, and K. Simonyan, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science (American Association for the Advancement of Science)*, vol. 362, no. 6419, pp. 1140–1144, Dec 2018.

[4] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA: The MIT Press, 2018.

[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv:1312.5602*, Dec. 2013.

[6] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv:1509.02971*, July 2019.

[7] J. Vitay. (2021, Feb.) Deep reinforcement learning - julien vitay. [Online]. Available: https://julien-vitay.net/deeprl/Introduction.html#sec:introduction

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv 1412.6572*, Mar 2015.

[9] S. Huang, N. Papernot, I. Goodfellow, Y. Duan, and P. Abbeel, "Adversarial attacks on neural network policies," *arXiv:1702.02284*, Feb 2017.

[10] I. V. N. Michèle Basseville, *Detection of Abrupt Changes: Theory and Application*. Prentice-Hall, Inc, April 1993, vol. 15.

Deep Reinforcement Learning for the Popular Game tag

Gustav von Knorring and August Söderlund

***Abstract*—Reinforcement learning can be compared to how humans learn – by interaction, which is the fundamental concept of this project. This paper aims to compare three different learning methods by creating two adversarial reinforcement learning models and simulate them in the game tag. The three fundamental learning methods are ordinary *Q-learning*, *Deep Q-learning* (DQN), and *Double Deep Q-learning* (DDQN).**

The models for ordinary Q-learning are built using a table and the models for both DQN and DDQN are constructed by using a Python module called TensorFlow. The environment is composed of a bounded square with two obstacles and two agents with adversarial objectives. The rewards are given primarily based on the distance between the agents.

By comparing the trained models it was established that only DDQN could solve the task well and generalize, whilst both the Q-model and DQN had more serious flaws. A comparison of the DDQN model against its average reward trends established that the model still improved regardless of the constant average reward.

Conclusively, DDQN is the appropriate choice for this adversarial problem whilst Q-learning and DQN should be avoided. Finally, a constant average reward can be caused by both agents improving at a similar rate rather than a stagnation in performance.

***Sammanfattning*—Förstärkande inlärning kan jämföras med sättet vi människor lär oss, genom interaktion, vilket är den fundamentala idéen med detta projekt. Syftet med denna rapport är att jämföra tre olika inlärningsmetoder genom att skapa två förstärkande motståndarinlärningsagenter och simulera dem i spelet kull. De tre fundamentala inlärningsmetoderna är *Q-learning*, *Deep Q-learning* (DQN) och *Double Deep Q-learning* (DDQN).**

Modellerna för vanlig Q-learning är konstruerade med hjälp av en tabell och modellerna för både DQN och DDQN är byggda med en Python modul, TensorFlow. Miljön är uppbyggd av en begränsad kvadrat med två hinder och två agenter med motsatta mål. Belöningarna ges baserat på avståndet mellan agenterna.

En jämförelse mellan de tränade modelerna visade på att enbart DDQN kunde spela bra och generalisera sig, medan både Q-modellen och DQN-modellen hade mer allvarliga problem. Genom en jämförelse för DDQN-modellerna och deras genomsnittliga belöning visade det sig att DDQN-modellen fortfarande förbättrade sig, oavsett det konstanta genomsnittet.

Sammanfattningsvis, DDQN är det bäst lämpade valet för denna motpart simulering medan vanlig Q-learning och DQN borde undvikas. Slutligen, ett konstant belöningsgenomsnitt orsakas av att agenterna förbättras i samma takt snarare än att de stagnerar i prestanda.

***Index Terms*—Reinforcement Learning, Neural Networks, Q-learning, Deep Q-learning, Double Deep Q-learning, Dual-agent Training.**

Supervisors: Erik Berglund

TRITA number: TRITA-EECS-EX-2021:153

I. Introduction

Historically, the initial intent of reinforcement learning was to induce "optimal control", which is a controller calibration in order to minimize certain aspects of a system. This purpose, combined with another predecessor, trial-and-error learning, merged in the late 1980s into what is currently known as modern reinforcement learning, according to [1].

Recent successful applications for deep reinforcement learning include numerous classical Atari games [2] and in the strategic board game of GO [3]. In just over half of the Atari games, the reinforcement learning agent accomplished a performance greater than that of a human as evidenced in [2]. One other project which is comparatively similar to this project is OpenAI's implementation for *hide and seek* [4], consisting of two hiders and two seekers. Their encouragement for future work included the proposal of a reduction in sample complexity and also adjusting the reward functions to better match with the predicted outcome. The reward function is explained further into the report. Apart from the implementation in popular video games, reinforcement learning currently serves a purpose in real-life scenarios such as the decision-making process of how and when to cool Google's servers and other diverse areas such as stock trading or healthcare, as demonstrated in [5]. The techniques in the reinforcement learning field are constructed to learn dynamically from experiencing a specific environment, instead of learning from a pre-made data-set.

The idea of this project is to simulate two reinforcement learning agents in a two-player game environment allowing them to learn the game by giving out different rewards and punishments for different actions, where they have the opposite objectives. Exploring this adversarial aspect of the project is important for future technological development since it allows for faster and better training of the reinforcement learning agents as explained in [6]. This particular project will use the game tag as its game environment where one agent will try to run for as long as possible whilst the other agent tries to tag them.

II. Theory

A. Reinforcement learning

The fundamental concept of reinforcement learning is largely similar to the way humans learn new elemental skills – by interacting with the surroundings through certain actions and experiencing the consequences [7]. According to [1], the algorithms behind reinforcement learning are implemented equivalently, where an agent takes actions in a simulated

environment and receives a reward based on the transition of the environment before and after the action was taken.

The so-called agent is the main acting part of reinforcement learning implementations. It decides which action to take through its current perception of the environment, where the environment consists of all other aspects apart from the agent, pointed out in [8]. Usually, the agent is only fed smaller pieces of information about the environment, so-called states, which can consist of arbitrarily many parameters, for example the "geographical coordinates of a robot" stated by Gosavi [9, p.3]. Besides the previously mentioned parties (i.e. the agent and the environment), reinforcement learning is composed of four other key components – policy, reward signal, value function, and a potential model, as illustrated in [8].

B. Markov decision process

Markov Decision Processes (MDP) is a key component of reinforcement learning. MDP is used to describe how the agent and the environment affect each other. The agent can alter the environment through its actions, consequently receiving rewards based on the value of the effects, according to [10]. The environment reacts and updates, which results in a new state of the environment that the agent has to interpret and take new actions upon. This process continues until terminated, usually when the final goal is achieved if there is one [8].

The fundamental mathematical description of an MDP is given by a tuple (S, A, T, p, r) where S corresponds to the state space, A corresponds to all possible actions, T is a set of time steps, p is a function which describes the probability of transitioning between two states, r corresponds to the reward function, as explained in [11].

According to [11], the main principle, mathematically formulated, for an MDP is that for each element in the set T, namely for each time step t, the agent performs an action $a_t \in A$ given the current state $s_t \in S$. The state then shifts into a new state $s_{t+1} \in S$ which gives the agent a reward r_t based on the consequence of the action. Discretizing the general MDP tuple into discrete time steps, described in [12], and using the γ-discounted criterion results in a tuple used in this report, namely (S, A, P, R, γ), with the main difference of the discount rate $\gamma \in [0, 1]$ is introduced, and the notation for the reward function and transition function being capitalized, similarly as in [9]. Each state in the state space is assumed to satisfy Markov Property meaning that the agent's taken action is solely based on the current state and not on previous states, as stated in [13]. This is mathematically written as

$$\begin{aligned} &P\{X_{t_n} = i_n | X_{t_{n-1}} = i_{n-1}, \cdots X_{t_1} = i_1, X_{t_0} = i_0\} \\ &\quad = P\{X_{t_n} = i_n | X_{t_{n-1}} = i_{n-1}\} \end{aligned} \tag{1}$$

as illustrated in [13], where X_t is the state at timestep t.

The policy in an MDP dictates how the agent will act in different states, according to [1]. For the current state s_t in the state space S, the policy π decides which action a_t in the action space A to take, mathematically denoted as $\pi : S \mapsto A$. The final objective of MDPs is to discover a policy that maximizes all the possible future rewards as defined in [14].

There are several policies, asserted in [15], where one of the three essential policies is epsilon-greedy. The epsilon-greedy policy uses a number $\varepsilon \in (0, 1)$ and a random number $\eta \in (0, 1)$ in order to determine which action to take. It is mathematically described by

$$a = \begin{cases} a_{rand} & \text{if } \eta \leq \varepsilon \\ \arg\max\limits_{a_i} Q_t(s_t, a_i) & \text{otherwise,} \end{cases} \tag{2}$$

where a_{rand} corresponds to a random choice between the set of actions, and where Q_t will be our action-value function. More about the action-value function further into the report. The number ε is a self-defined variable where a smaller value results in the policy choosing the optimal action more often and a larger value results in a more randomized selection.

In order for the policy to calculate the potential future rewards, the discounted return function is introduced, according to [8]

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}, \tag{3}$$

where R_{t+k+1} denotes the reward given in state k time steps in the future. Increasing the discount factor γ results in the policy prioritizing future rewards higher compared to decreasing it. The discounted return function is primarily used to define the state-value function for policy π

$$v_\pi(s) = E_\pi[G_t | S_t = s] \tag{4}$$

which describes the expected value of the agent being located in the current state. But it is also used in the definition of the action-value function for policy π

$$q_\pi(s, a) = E_\pi[G_t | S_t = s, A_t = a] \tag{5}$$

which describes the anticipated value of the agent performing the action a in its current state s and continuing obeying the policy π. [8]

C. Q-learning

In the simplest form, Q-learning utilizes a Q-table, a table with actions as columns and states as rows, represented as a function $Q\colon S \times A \to \mathbb{R}$ where S is a discrete set of states and A is a discrete set of actions. Each element in the table represents the "quality" of taking the given action in the respective state. The elements in the table are iteratively updated, "learned", based on the experienced reward of the action, previous rewards, and expected future reward. For finite Markov decision processes, Q-learning can be used to find an optimal action-selection policy as [16] proves convergence of the algorithm. The MDP theory serves as a ground for these methods.

Before the iterative process begins the function Q is initialized with arbitrary values (high or low) chosen by the programmer. Then at time t, at the state s_t, the agent selects an action a_t, either at random (exploration) or by the maximal Q-value (exploitation). After the action, the agent observes a

reward r_t and is presented a new state, s_{t+1}. The Q function is updated as such

$$Q_{\text{new}}(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \cdot [r_t + \gamma \cdot \max_a(Q(s_{t+1}, a)) - Q(s_t, a_t)] \quad (6)$$

where α is the learning rate between 0 and 1, and γ is the discount factor. The learning rate determines how important new information is compared to old information, and how much of the old information should be kept or forgotten in the Q-table, according to [15].

Using different initial conditions can cause certain behaviors, for example initializing the Q-table with high values and when the action is taken for the first time the corresponding table element is only set to the reward. This encourages exploring without using random actions as each never before taken action is likely to be picked. After the first time an action is picked, the algorithm continues normally as described.

D. Deep Q-learning

Q-learning has a couple of disadvantages. For example, it only works for discrete states and actions. Further due to physical computer limits, the algorithm falters for large amounts of actions and states as the time to explore becomes very long and the probability of visiting a certain state is very low, and eventually the memory consumption becomes unmanageable.

To circumvent these issues the Q function can be replaced with a function approximation, then the algorithm can be applied to both larger and continuous problems. One such function approximator used is artificial neural networks, the method then becomes Deep Q-learning (DQN).

1) Artificial neural networks: According to [17] artificial neural networks are universal function approximators that can approximate any given continuous function adequately with a sufficient number of nodes in the network. The input and the output of the network is a fixed-length vector in any vector space.

Generally, the nodes, or neurons, in the network are structured into ordered layers, where each node is connected to each node in the previous layer. Each connection has an associated weight. This setup is called a feed-forward network.

Aside from the input layer, each node in the network obtains a value by adding a threshold value to a weighted sum of the output of the previous node-layer before applying a non-linear activation function which yields the new value for the node used by the next node-layer in the network.

Mathematically the node takes the value $f(b + \sum_{i=1}^{n} x_i w_i)$ where x_i is the input values from the previous nodes, w_i is the corresponding weight, n is the number of previous nodes, and b is the threshold value (also called bias), f is the activation function. That process continues for each layer until each output node has an assigned value and the output vector can be read from the output layer, i.e. the nodes of the last layer.

Any continuous function can be approximated by using enough nodes and adjusting the threshold values and weights of the network. The weights are usually initialized randomly and are later updated progressively, in small steps, using an algorithm called gradient descent on a so-called loss function. When the loss function is minimized the network is deemed to perform well. This report specifically uses the ADAM algorithm [18].

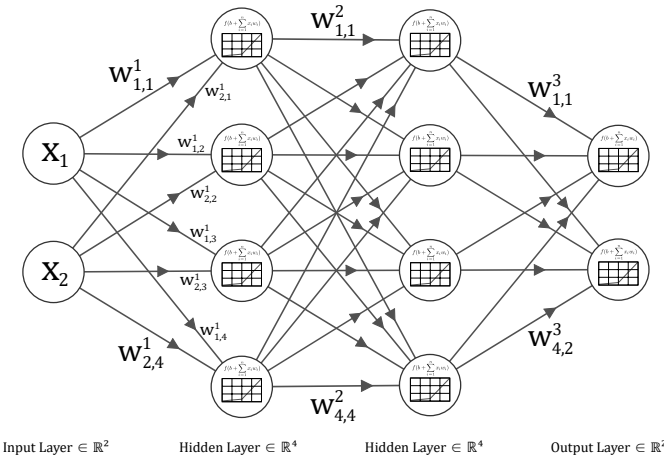


Fig. 1. Visual representation of a neural network with two hidden layers between the input and output layers. The weights are represented as arrows as well as the variables $\mathbf{w}$ with indices (top index is layer, bottom index is [from node], [to node]). An activation function is also visually represented inside the nodes.

Artificial neural networks can sometimes generalize earlier experiences, and in Deep Q-learning yield good actions in previously unseen states and policies, as illustrated in [2] and [19].

2) Artificial neural networks as a Q-function: In Q-learning case the Q-function is parameterized by the values in the Q-table, however, when using a neural network as Q-function the function is parameterized by the network, weights and biases. Therefore equation (6) does not apply. Let those parameters be called $\boldsymbol{\theta}$. Following the previous formulation, we effectively want to minimize the difference between our target $Q^{\text{target}}_{\boldsymbol{\theta}}(s_t, a_t) = r_t + \gamma \cdot \max_a(Q_{\boldsymbol{\theta}}(s_{t+1}, a)$ and current $Q_{\boldsymbol{\theta}}(s_t, a_t)$, since that would result in $Q_{\boldsymbol{\theta}}(s_t, a_t)$ coming closer to our target. We create the loss function $L(\boldsymbol{\theta})$, as subject to minimization.

$$L(\boldsymbol{\theta}) = \sum_{i \in T} (Q_{\boldsymbol{\theta}}(s_t, a_t) - Q^{\text{target}}_{\boldsymbol{\theta}}(s_t, a_t))^2 \quad (7)$$

The basic method of minimizing this function is the gradient descent method

$$\boldsymbol{\theta}_{\text{new}} \leftarrow \boldsymbol{\theta} - \alpha \frac{\partial L}{\partial \boldsymbol{\theta}} \quad (8)$$

where α is a constant, and this is the basis of the ADAM algorithm [18] we use. In reality, we do not sum all states and actions in T, instead the gradient is approximated by only using a smaller sample, a batch, of random states and actions that have recently occurred. Using this approximation for the loss function gradient, the minimization method becomes stochastic gradient descent. Often, a regularization term $\lambda \cdot ||\boldsymbol{\theta}||^2$ is added to L to prevent large weights in the network, which helps generalization [20].

3) Double Deep Q-learning: According to [21], in noisy environments or action-value approximations, such as a neural

network, Q-learning may overestimate the ”quality” of an action, due to the future maximum reward being approximated with the same Q-function as the current policy. This issue can be corrected using two separate Q-functions, Q_a and Q_b. Each Q-function uses the other Q-function for the future approximation term and is usually trained simultaneously in a symmetric fashion.

This can be combined with Deep Q-learning, resulting in Double Deep Q-learning (DDQN), to achieve better results.

III. Method

A. Environment

The environment used for this paper is a bounded quadratic area of continuous space containing two fixed obstacles. Furthermore, the regions that the obstacles span are illicit terrain for the agents to explore. The agents themselves are initialized onto the environment at random valid locations and can move inside the environment in a discrete set of directions with constraints on its maximum velocity in each direction.

The room is 20 by 20 agent diameters and the max velocity is 0.2 agent diameters per game-tick, where a game-tick is simply the time it takes for both agent algorithms to pick an action.

A visual representation of the environment can be done by using the Python module TkInter which enables a demonstration of either two trained models or one human player playing against a model. Figure 2 shows an initial state of the environment where the red circle corresponds to the agent who wants to tag, and the green circle corresponds to the agent who wants to escape.

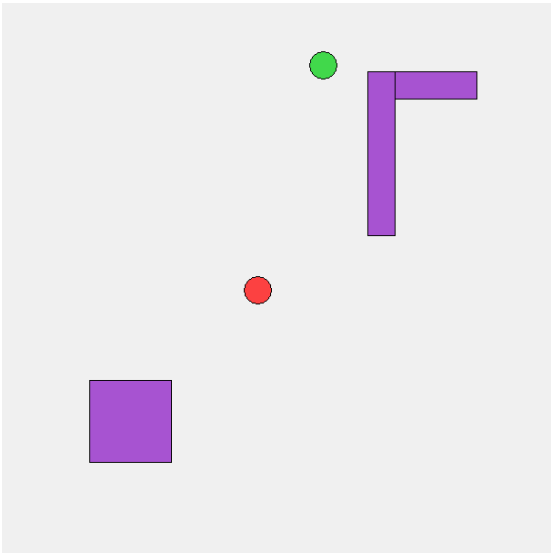


Fig. 2. Visual representation of the environment with two obstacles shown in purple and the two agents. The hunting agent (agent A) shown in red and the escaping agent (agent B) shown in green.

The two agents have adversarial objectives, thus are rewarded contrarily, like a zero-sum game. When one agent receives a positive reward, the other agent receives an equal but negative reward. The rewards are primarily based on the distance between the agents, given that they are within each other's field of vision, calculated with an inverse of the distance between the agents' midpoints. Otherwise, when the game is over, i.e. the green player was tagged, the agents will receive a final reward based on the results. This final reward is larger than the sum of all discounted potential future rewards described in equation (3).

The states that are presented to the agents contain information about the opposing agent's last seen relative position, and the field of vision. The field of vision consists of several equally spaced rays extruding in evenly spaced directions up to a defined length of sight for the agents. The directions of the rays are the same as the possible movement directions for the agents. The length of the rays is shortened if obstacles and walls exist within its vision. Finally, the states also include information about whether the opposing agent is within its field of vision.

B. Learning methods

1) Q-learning: For ordinary Q-learning, the different states of the environment are discretized in order to reduce the maximum size of the Q-table and allow for faster exploration. Firstly, the number of vision-rays is set to eight, and their lengths are discretized into three steps, one step for lengths between $[0, 2)$ agent diameters, another step for lengths between $[2, 4)$ agent diameters, and the last step for lengths between $[4, 6]$ agent diameters. The maximum sight length was chosen to six agent diameters after some trial-and-error trying to balance the Q-table's size while retaining sufficient information. This length is also used in the other two learning methods. Information about the opposing agent's relative position is discretized based on which circle sector between rays the opponent is currently in and the distance between the agents. This discretizing function returns a unique hash value for each different state. The actual Q-model is implemented through a dictionary with the unique hash value as a key and a list of Q-values for each action as the dictionary's value.

The training stage is initialized with an $\varepsilon = 0.99999$ such that the model is less biased to take the optimal action according to its own predictions and chooses random actions in the early stages of training. After each played game ε is reduced by a factor $\varepsilon_{decay} = 0.999996$ to make the model increasingly biased towards taking the best action for each state. The training is also initialized with a learning rate, $\alpha = 0.15$, and a discount factor, $\gamma = 0.99$. The number of games played for training the Q-learning model is $5\,000\,000$ games. The value of ε and ε_{decay} is chosen arbitrarily close to 1 such that the agent will explore much and then less and less during the training. The decay continues until ε is 0.005. The value of γ and α was chosen after some trial-and-error trying to get a bit better performance, especially a γ close to 1 helped in our case which promotes a more long term strategy for the agents. For all algorithms, namely Algorithm 1, 2 and 3 the number of games is equivalent to the number of episodes.

The algorithm for Q-learning is presented in Algorithm 1. This is performed for both agents synchronously.

2) Deep Q-learning: The model creation for Deep Q-learning is done through predefined Python modules called

Algorithm 1 Q-learning with epsilon-greedy policy

Initialize Q-table for agents, with all elements initially set to 0.
for each episode **do**
  for each timestep, t, in game time **do**
    Get current discretized states for agents.
    Pick an action according to epsilon-greedy-policy, see equation (2).
    Perform the action and store the state transition (S_t, A_t, S_{t+1}, R_t).
    Update the Q-table values based on the transition according to equation (6).
    if terminal state **then**
      Break, exit loop.
    end if
  end for
  Reset environment with new initial state.
  If non-constant epsilon, update epsilon value.
end for

Algorithm 2 Deep Q-learning

Initialize the neural network with random weights and biases according to tensorflows default distribution, and with a regularization factor of $5 \cdot 10^{-6}$, and with a LeakyReLU activation function.
Initialize the replay memory (a deque).
for each episode **do**
  for each timestep, t, in game time **do**
    Predict the best action given the current state using the neural network and the policy to select action for the agent.
    Perform the chosen action.
    Save the state transition (S_t, A_t, S_{t+1}, R_t) to replay memory.
    Train the network, according to ADAM with the previous described loss function (7), on a random batch from the replay memory.
  end for
  Reset environment with new initial state.
  If non-constant epsilon, update epsilon value.
end for

TensorFlow and Keras. The actual model contains five layers (one input layer, one output layer and three hidden layers in between). The input layer contains nodes equal to the number of inputs for the states and the last output layer contains nodes equal to the number of different actions the agents' can perform. The first and second hidden layers contain nodes equal to twice the input size and, the last hidden layer contains nodes equal to the mean of the number of nodes in the second hidden layer and the output layer.

The model is then used for training. Training the Deep Q-learning network is structurally done similarly to ordinary Q-learning. The main difference is that instead of representing the model with a dictionary the model for Deep Q-learning is represented by an artificial neural network. For the training, it samples a random batch of 128 transition tuples, (s_t, a_t, r_t, s_{t+1}), from its replay memory and trains against that by updating the network parameters $\boldsymbol{\theta}$ with a stochastic gradient descent algorithm, in this case ADAM [18].

The training is initialized with a predefined set of variables that are relevant for Deep Q-learning. The discount factor, $\gamma = 0.99$, is set to a relatively high value in order to make the model account for future rewards more favorably. The learning rate parameters are set by the default values in TensorFlow for ADAM algorithm [18]. $\varepsilon = 0.99$ corresponds to a relatively high exploration rate and after each game it is scaled down by a factor $\varepsilon_{decay} = 0.995$ in order to make the model exploit its previous knowledge further as the training proceeds. The number of games played during training is about $700\,000$.

The algorithm for Deep Q-learning is presented in Algorithm 2. The two agents act synchronously, and due to performance reasons our code actually plays a number of games at the same time, which are not reflected in the schematized algorithm.

3) Double Deep Q-learning: For Double Deep Q-learning, the model creation is very similar to the one for Deep Q-learning, however, two models are created for each agent instead of only one model. The primary model is trained much more often than the secondary model. The primary model's training procedure is identical to the one for Deep Q-learning and the secondary model is trained once for every ten times the primary model is trained.

Training the secondary model consists of replacing the current weights with a weighted average of the primary model's weights and the current secondary weights. The weight of the primary model weights is $\tau = 0.125$ so it is always behind except when 600 training steps have passed then $\tau = 0.999$. This causes the secondary model to be updated in small steps, and every 600 training steps almost fully catching up to the primary model. These values were chosen arbitrarily to achieve this behavior.

The algorithm for Double Deep Q-learning is presented in Algorithm 3. Similar to Deep Q-learning (DQN), this is performed for multiple games where each agent is acting synchronously.

C. Evaluation of the models

After training, the models were tested and evaluated in five standardized tests. Test 1 (T1) is a "surround test" where the location for the escaping agent is statically set in 16 angles and 5 different lengths from the hunting agent. The angles are evenly spread throughout a circle and the distances are defined as the agent's unit length times $[2, 3, 4, 4.75]$ and the final distance is defined as 120 percent of the agent's sight length. The model receives a point if the hunting agent successfully tags the escaping agent.

Test 2 (T2) is a "search test" where the location for the hunting agent is set at 9 predefined locations and the location for the escaping agent is set identically for all tests. The model receives a point if the hunting agent tags the escaping agent.

Test 3 (T3) is a "escape test" where the hunting agent travels at a constant speed along a line in the playfield. The model receives 0.1 points if the escaping agent successfully flees.

Algorithm 3 Double Deep Q-learning

Initialize the primary neural network (as in DQN).
Initialize the secondary neural network (as in DQN).
Initialize the replay memory (a deque).
for each episode **do**
  for each timestep, t, in game time **do**
    Predict the best action given the current state using the primary neural network and the policy to select action for the agent.
    Perform the chosen action.
    Save the state transition (S_t, A_t, S_{t+1}, R_t) to replay memory.
    Train the network, according to ADAM with the previous described loss function (7), on a random batch from the replay memory, using the secondary neural network for predicting future rewards.
    if a certain number of steps has passed, 600 **then**
      Update the secondary model weights to be very close or identical to the primary model as described in the previous paragraph.
    end if
  end for
  Reset environment with new initial state
  If non-constant epsilon, update epsilon value.
end for

Test 4 (T4) is a "new environment test" where a new environment is defined as, simply, two obstacles forming a corridor which the agents were never shown during training. The relevant actions are left and right and the location for the agents is set randomly inside the corridor, but still within each other's field of vision. The escaping agent remains stationary and the model receives a point if the hunting agent successfully tags the escaping agent.

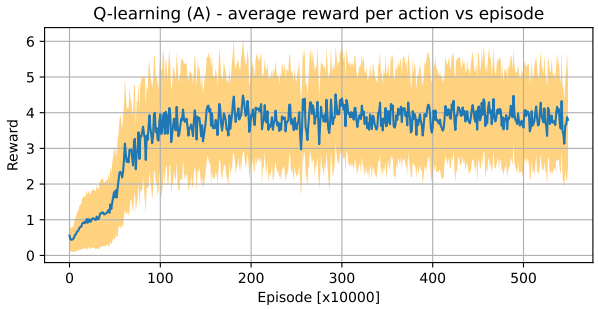
Test 5 (T5) is also a similar "new environment test" to T4. It has a similar environment and similar locations for the agents. However, in this test, the model receives a point if the escaping agent successfully flees from the hunting agent that moves at a slow constant speed along the corridor.

IV. Results

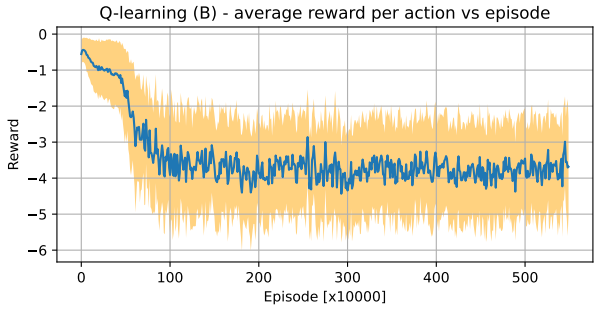
A. Rewards

1) Q-learning: The figures, Figure 3a and Figure 3b, represent the average rewards that the escaping agent and the hunting agent received during each game with ordinary Q-learning implemented. The average rewards are defined as the total rewards received for each episode divided by the number of actions taken in the episode, this applies to all three learning types. The line represents an averaging of the 10000 nearest points to allow for easier recognition of trends.

The loss was also calculated and plotted in Figure 4a and Figure 4b. The loss is defined as the squared difference between the target Q-value and the best Q-value in the agent's current state. Similarly to previously, the average is calculated for every 10000 points.

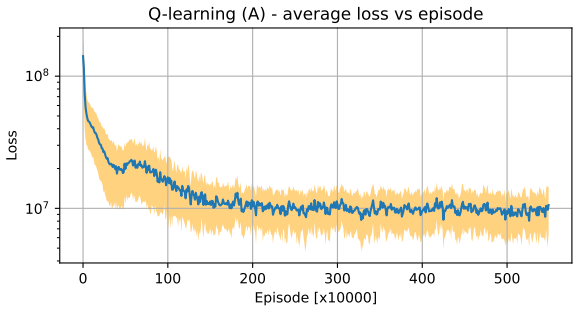


(a) Agent A

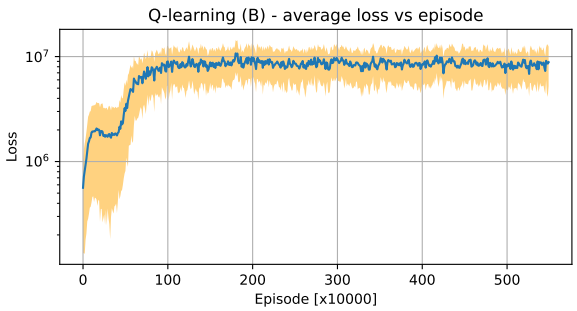


(b) Agent B

Fig. 3. The agents' average reward per action received for each separate game during training procedure with Q-learning implemented. (a) Represents the average rewards for the hunting agent, (b) represents the average rewards for the escaping agent. The yellow shaded area represent one sample standard deviation.



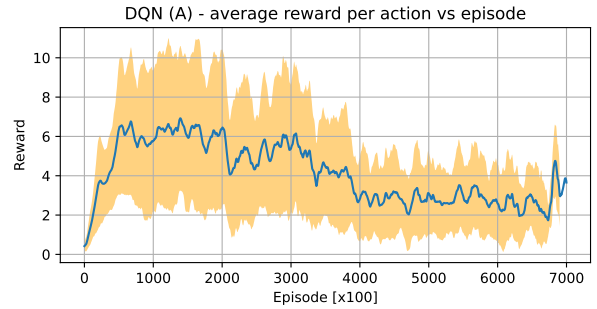
(a) Agent A



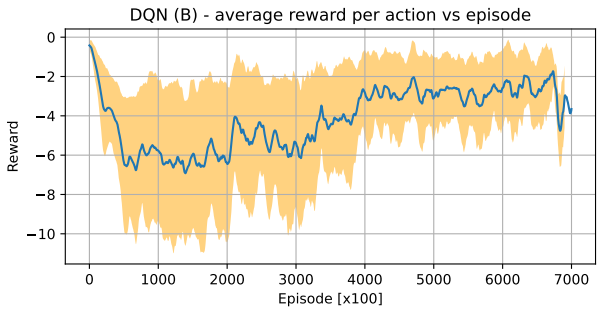
(b) Agent B

Fig. 4. The agents' loss that it accumulated for each separate game during training procedure with Q-learning implemented. (a) Represents the loss for the hunting agent, (b) represents the loss for the escaping agent. The yellow shaded area represent one sample standard deviation.

2) Deep Q-learning: Similar to ordinary Q-learning, in order to receive easier interpretable trends the average of the 100 nearest points was calculated. The figures, Figure 5a and Figure 5b, represent the average rewards received in each game with Deep Q-learning implemented.

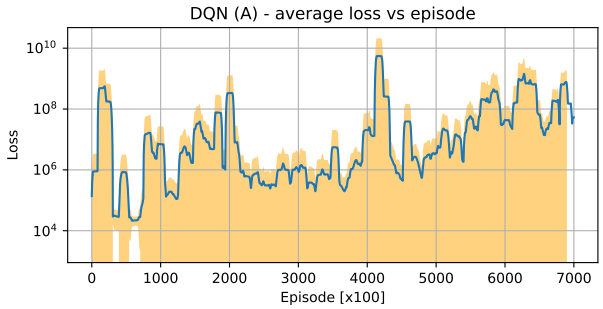


(a) Agent A

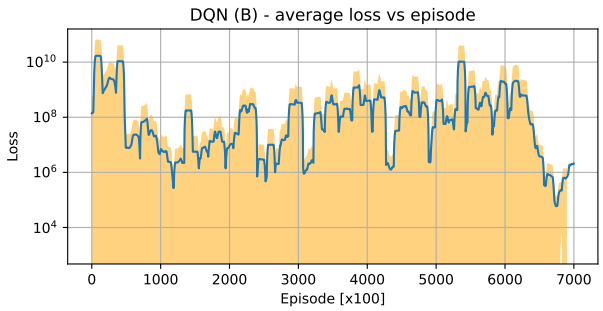


(b) Agent B

Fig. 5. The agents' average reward per action received for each separate game during training procedure with Deep Q-learning implemented. (a) Represents the average rewards for the hunting agent, (b) represents the average rewards for the escaping agent. The yellow shaded area represent one sample standard deviation.



(a) Agent A



(b) Agent B

Fig. 6. The agents' loss that it accumulated for each separate game during training procedure with Deep Q-learning implemented. (a) Represents the loss for the hunting agent, (b) represents the loss for the escaping agent. The yellow shaded area represent one sample standard deviation.

The loss is also calculated and plotted for Deep Q-learning. Similarly to Q-learning, it is calculated as the squared difference between the predicted outcome and the actual outcome. This is done automatically by configuring our model loss and using Keras' function *train_on_batch*. The figures, Figure 6a and Figure 6b, represent the losses acquired during training for the Deep Q-learning model.

3) Double Deep Q-learning: The figures, Figure 7a and Figure 7b, represent the total rewards for each game for the primary model in Double Deep Q-learning. Similar to previous results, averaging of the 100 nearest points is performed for a smoother line and allows for easier recognition of trends.

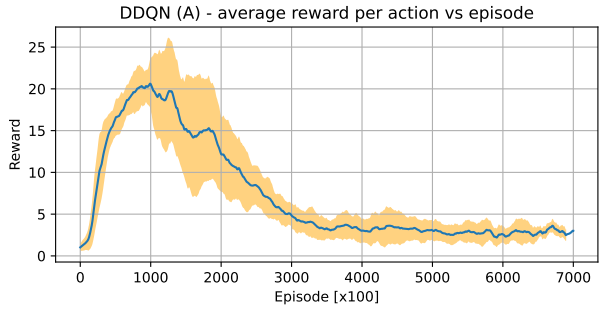
Finally, the loss was also calculated for the Double Deep Q-learning model. This is calculated identically to Deep Q-learning as explained previously. The figures, Figure 8a and Figure 8b, represent the losses acquired during training for the Double Deep Q-learning model.

B. Performance

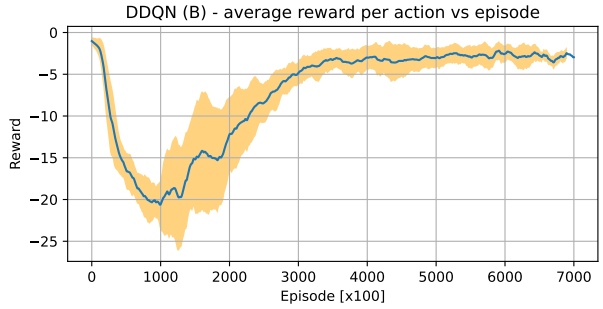
Table I shows the results for the models' average performance for standardized tests, T1 through T5, as defined in section III-C.

TABLE I
PERFORMANCE FOR STANDARDIZED TESTS

	T1	T2	T3	T4	T5	Sum
Q	66.38	24.13	11.06	1.00	0.00	102.57
DQN	5.13	3.50	8.56	0.50	3.50	21.19
DDQN	62.63	28.38	13.45	6.25	7.75	118.46

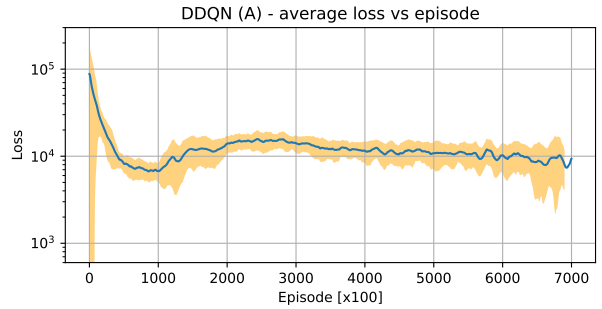


(a) Agent A

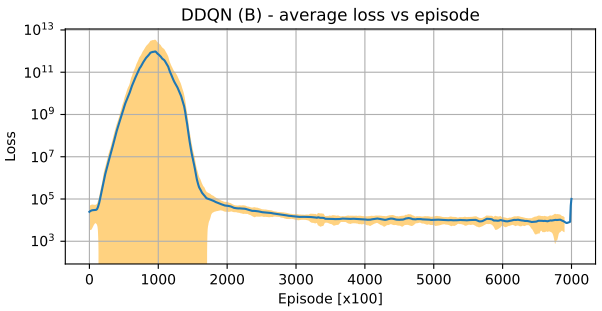


(b) Agent B

Fig. 7. The agents' average reward per action received for each separate game during training procedure with Double Deep Q-learning implemented. (a) Represents the average rewards for the hunting agent, (b) represents the average rewards for the escaping agent. The yellow shaded area represent one sample standard deviation.



(a) Agent A



(b) Agent B

Fig. 8. The agents' loss that it accumulated for each separate game during training procedure with Double Deep Q-learning implemented. (a) Represents the loss for the hunting agent, (b) represents the loss for the escaping agent. The yellow shaded area represent one sample standard deviation.

V. Discussion

A. Evaluation of results

1) Q-learning: For ordinary Q-learning it is evident, according to Figure 3a and Figure 3b, that the average rewards approaches to a seemingly constant value. The error margin is relatively low, which indicates that many of the trained models have a similar reward average. Furthermore, the loss also approaches a constant value, but in opposite directions, according to Figure 4a and Figure 4b. This means that the model's Q-values are updated at a significantly larger rate in the beginning compared to later in the game, for agent A, and vice versa for agent B. For episodes after 200 [x10000], as seen in Figure 4a, the loss remains at a seemingly constant level and the rewards have also flattened. This may indicate that the model does not improve and has reached its maximum potential. Figure 4b shows a similar behavior but initially the model does not update its Q-values considerably, but attains a similar constant level after 100 [x10000] episodes. This initial low update frequency is presumably caused by the fact that agent B's initial action is to remain stationary, which initially is the optimal action since agent A does similarly. Because of agent B's action to remain stationary, it does not need to update its Q-values remarkably, thus the loss initiates at a low level for agent B. However, as agent A starts to receive more rewards, possibly because it tags agent B more frequently, agent B needs to find a better strategy than remaining stationary. This causes the losses to increase because it has to update its Q-values more frequently.

While also comparing Figure 3a and Figure 3b, it is clear that they are mirrored around the x-axis. This is due to the fact that the agents receive opposite rewards due to a zero-sum game. Since agent A receives positive rewards and agent B receive negative rewards one might believe that agent A is the best agent. However, due to the formulation of our rewards where agent A are generally more likely to receive higher rewards, this may alter in practice. Since the agents are trained against each other, they can only adapt to their opponent's strategies and possibly outperform them. This does not mean that the agents will outperform a human. For example, if agent B follows a strategy where it is constantly stationary, agent A will eventually find a strategy that simply travels towards agent B, thus finally wins the game. If this strategy is used against an actual human, the human will most likely win since they can evade the hunting agent simply by traveling orthogonally compared to the hunting agent's direction of travel.

Finally, the formulation of the states used in this paper can cause unexpected behavior. Since the models are only allowed to perceive the distance to walls and obstacles and the other agent's position and not the other agent's direction of travel, the action that it chooses may not be optimal. For example, if both agent A and agent B travel left and reach a state they both have observed previously, the optimal action may have been to travel right, according to its previous experience. This may cause the agent to take that action and lose sight of the other agent. Otherwise, to travel left may become the new optimal action, but that may cause the same situation if both agents travel up or down.

2) Deep Q-learning: Identically to ordinary Q-learning, the achieved rewards are mirror images of each other, as seen in Figure 5a and Figure 5b. However, one difference between ordinary Q-learning and Deep Q-learning is that Deep Q-learning does not have equally clear trends. The rewards for agent A does initially increase, and then decrease at a lower rate, but it has a more stochastic behavior than ordinary Q-learning. An interpretation of the initial trend is that agent A improves at a higher rate compared to agent B, thus receiving higher rewards for future episodes. However, for episodes after 2000 [x100] the rewards decreases, for agent A, which may indicate that agent B slowly adapts to agent A's possible strategies. Furthermore, for episodes after 4000 [x100] the decrease stagnates and it attains a rather randomized pattern, with a notably constant average value. This may indicate that neither of the models improves nor deteriorates. However, the peaks and valleys suggest that the models improve locally, only to be outmatched by the other agent's local advancements. This may indicate that the models became slightly better as time progresses, but is not visible in the average rewards. Finally, the average rewards jump rapidly at around 6700 [x100] episodes which may signify that agent A adopted another better strategy that may outperform agent B. Conclusively, the rewards for Deep Q-learning have a much more stochastic behavior compared to ordinary Q-learning. This is also supported by the larger error margin, also seen in Figure 5a and Figure 5b. Another reason for the stochastic behavior might be overestimations causing the model to sometimes deteriorate as it wrongly assumes actions to be better than they actually are causing a large loss.

The losses for the Deep Q-learning models are significantly different from the losses for ordinary Q-learning. This is

shown in Figure 6a and Figure 6b. The losses attain a noticeably larger value at certain episodes and appear to resemble a smaller city's skyline. One plausible explanation for this update-peaks may be that the agent lost those particular games, thus needing to update the model's values which consequently resulted in a massive peak. Whilst the average rewards appeared to slowly approach a constant value, the losses do not. With this being said, it does not necessarily mean that the models do not improve, due to the adversarial formulation of this problem. Since the agents have opposite objectives it is possible that they both improve at a similar rate, which would induce this behavior for both the average rewards and the losses.

Whilst evaluating the models we saw that it was significantly better at generalizing compared to the models for ordinary Q-learning. This is seen in Table I for T4 and T5. The model adapts easier to new environments. This suggest the neural network, which is a general function approximation, has learned some intuition to a general game of tag. Ordinary Q-learning utilizes a table of Q-values for each state and action experienced, which means that the Q-learning model has no knowledge of never before seen states. However Q-learning still got one point in T4, this is due to the agent being placed in such a way that the new state was discretized such that it was mapped to a state it recognized.

3) Double Deep Q-learning: According to Figure 7a and Figure 7b, the average rewards that the agents receive during practice approaches a small constant value for episodes larger than 3000 [x100]. However, the rewards that agent A receives initially increases until episode 1000 [x100], only to be decreased until around episode 3000 [x100]. This may be caused by agent A adopting a relatively good strategy that agent B does not adapt to until its peak at episode 1000 [x100]. The decline may then correspond to agent B slowly becoming better at recognizing agent A's strategy, thus outperforming it. For episodes larger than 3000 [x100], the constant value may be explained by the fact that both agents learn at a similar rate. The small peaks for episodes 3700, 4400 and 4800 [x100], as seen in Figure 7a, are probably caused by agent A learning a better strategy, which would cause it to receive higher rewards, only for agent B to find an even better strategy, which levels the play-field once again. Even if the average rewards are rather constant, that does not mean the agents do not improve but improve at similar rates. Finally, it is also visible, similarly to both ordinary Q-learning and Deep Q-learning, that the average rewards are mirror-images of each other. This mirror-like behavior is, once again, expected due to the zero-sum game formulation.

The losses for agent A, as seen in Figure 8a, seem to slowly decrease, and may potentially decrease for future episodes, meaning that the model may not have reached its full potential. However, due to lack of time, the training had to be stopped at around 7000 [x100] episodes. The loss for agent B has a remarkably large peak at episode 500 [x100], as seen in Figure 8b. However, it quickly returns back down to a seemingly constant value. This spike may be caused by agent B needing to update its values significantly in order to match agent A's performance.

Whilst evaluating the Double Deep Q-learning models it was determined that they are noticeably the best, compared to ordinary Q-learning and Deep Q-learning, as seen for nearly all tests in Table I. It is significantly better at generalizing, as seen for T4 and T5, and it is similar in performance as ordinary Q-learning for the other tests. The primary model is constructed identically to Deep Q-learning, but with the introduction of a secondary model, the primary model performs better. This may be caused by the primary model using the secondary model to predict future Q-values, which would reduce the overestimation of the Q-values, thus resulting in better performance. It reduces the overestimation because it uses two slightly different separate models to predict good actions and for predictions of future reward instead of just a single model, as in [21]. As also stated in [21], not only do noise in environments cause overestimations, but also the usage of action-value estimates causes overestimations. This project confirms that claim to a certain degree. The environment used in this paper is, arguably, noise-free because it has a relatively small finite set of states. Although, even in this noise-free environment it appears that the model for Double Deep Q-learning performs the best. The fact that the model for Double Deep Q-learning performs the best can also be supported by its action-value approximation. It appears that models using themselves to predict future Q-values are eager and overestimates the values, but models using a secondary model (i.e. a previous version of itself) are uneager and approximates future Q-values more correctly.

B. Behavior of the models

The models were also tested against a human in order to observe any possible reasons to the performance, as seen in Table I. The ordinary Q-learning model acted rather primitively. It was not sufficient in tagging nor searching for the human player. This is possibly due to the fact that the model has not been trained against another sufficient model. The Q-model was only trained against another Q-model, meaning that they can only become better than each other. A human has much better generalizing capabilities, thus outperforming the Q-model massively. Furthermore, testing against a human may result in states previously unseen, which would be similar to T4 and T5 in Table I. However, whilst only observing the models play against each other it was noted that agent A was greater than agent B, which may have been the cause of the performance results as seen for T1, T2 and T3 in Table I.

The Deep Q-learning models were significantly better at playing against a human. Similarly, this is because any test against a human would most likely classify as T4 or T5 in Table I, where indeed the Deep Q-learning model is better than the ordinary Q-model. However, an observation was made on their tagging performance. The Deep Q-learning models were regularly unable to tag the opposing agent, whether it was a human or another deep Q-model. If the other player remained stationary the model did not tag the opponent, instead it altered between two locations at a constant distance from the opponent. This flaw may have been the reason for the lack of performance as seen for T1, T2 and T3 in Table I.

Finally, the Double Deep Q-learning models performed according to its claimed performance in Table I. It generalized the best whilst playing against a human, which is supported by its performance according to T4 and T5 in Table I, and it matched the Q-learning model in the other tests. These results and observations show that the model for Double Deep Q-learning performs generally the best for either known environments or for new environments, which is also confirmed by the sum of the tests in Table I. However, the worst performing model is the deep Q-model, because it could not perform the task successfully, only jumping between two positions instead of tagging the opponent. Lastly, the ordinary Q-learning model did not perform well against a human or for generalized situations, but it did outperform its opponent consistently. However, in order to make a complete conclusion about the models' performance, it would be necessary to simulate the game using two different learning methods that represent each agent in the same simulation.

C. Difficulties of implementation

During the initial stages of training the neural network models and the ordinary Q-learning model we quickly discovered the rise of a seemingly bad strategy for the escaping agent. It regularly decided to travel towards corners, especially the bottom right corner. Furthermore, the hunting agent also had some difficulties capturing the escaping agent. When within range with the escaping agent trapped in a corner, the hunting agent commonly stayed within a fixed distance without capturing the trapped agent. All of these aspects above led us to believe that primarily the reward functions were faulty, hence we altered them slightly which reduced the problem sufficiently for the results to be more accurate.

Finally, whilst testing the models we also discovered, for ordinary Q-learning, that both agents occasionally decided to remain stationary which we concluded was due to insufficient training episodes. This problem was solved by training the model for a longer period of time. We also thought that perhaps the deterministic nature of the agent policy made it difficult for the agent to learn and/or take proper actions. We have not been able to test this due to time constraints.

Another difficulties we encountered during the implementation were mismatching support between our Python version and the modules we used, by downgrading Python version the errors were resolved.

When we first were doing tests on the training, we found it to be unexpectedly slow. As it turns out, TensorFlow is optimized for larger loads of data. By instead doing predictions for 128 games in parallel instead of a single game, the overall average training time got significantly faster.

However, our training and gameplay were still quite slow. Initially, we thought that putting the model on the GPU would make it faster due to its parallelizable nature. However, both our model and states were too small to effectively take advantage of the grand parallelization, resulting in various stalls due to communication between CPU and GPU and perhaps other slowdowns that we did not understand. Profiling showed that using a GPU was not faster than CPU. Perhaps if we were proficient in the CUDA language we could have made the game entirely for GPU and then optimally take advantage of the GPU functionalities.

Next up we thought about using PyPy, since it performs significantly faster on various tasks [22]. Testing our code with PyPy resulted in a ten times speedup. However, not all of TensorFlow was compatible with PyPy so we were forced to stay with the default slower CPython interpreter.

Another smaller issue we saw while training was that as days went by the TensorFlow process RAM usage got larger and larger. Initially, in the first hours, the process requires a moderately 200-300 MB, but after three days it consumes above 5 000 MB. This did not occur with our Q-learning program. This issue suggests that there might be a memory leak in TensorFlow, however, TensorFlow is a very commonly used module and it would be unlikely that would be the case. Instead, it might be an issue with the certain version we use, or our computer setup, or we are perhaps using TensorFlow wrongly, or other reasons unknown to us.

During the training, there were some interruptions causing discontinued training, for example out of memory error, power outage, and maintenance downtime. However, backups were made frequently that could be booted from aiding those disruptions.

D. Future work

Our primary suggestion for future projects in the same subject is to test and compare more than three different types of learning methods. This might be interesting since it will give a more comprehensive comparison regarding which is the optimal learning method for this type of problem. Furthermore, for all of the newly tested learning methods, it might also be interesting to analyze different policies. We chose to exclusively analyze the epsilon-greedy policy, which may not be the best possible policy for this problem, thus other policies may be interesting to analyze.

Secondly, our environment was not randomly built each time the game starts. One could experiment with using random obstacles in the environment similar to [4].

Thirdly, we believe it is important for future projects to analyze the robustness of these learning methods by either adding a small random noise to each state either during training or during evaluation or produce noise that optimally perturbs the evaluation of actions. Since this project utilizes a continuous environment it is a closer representation of real-life two-dimensional scenarios than a discrete space is, thus it can form a good foundation for the decision of which learning method should be used in physical applications. The robustness of the learning methods are therefore important to research since real-life scenarios generally contain a more stochastic behavior, thus making robust models a necessity.

Another perspective on robustness is to let the agents randomly play against a past version of the opponent. In this project, the agent only plays against the current opponent's strategy. A more general strategy might be achieved if the models were trained against previous strategies as well, and might yield performance gains with less training as different strategy aspects have to be met earlier in the training process.

Furthermore, this project was written and implemented solely in the programming language of Python which cannot compare to, for example, C or C++ when it comes to computing speed. It may be interesting to implement the same project or a similar project in C++ and compare the difference in computing speeds. Also, C or C++ is usually the programming language of choice when implementing computer programs in embedded systems, as stated in [23], which would make this project more applicable for simulating using real-life robots in a controlled environment.

It would also be interesting to study how the different models would behave training and playing against each other, for example a Double Deep Q-learning model against an ordinary Q-learning model.

Lastly, it would be interesting to see if a large number of pre-prepared state-transition-tuples could be used to pre-train the models before exposure to the actual environment and what kind of benefits that might give.

VI. Conclusion

The conclusions that can be drawn from the results and discussion are:

- Ordinary Q-learning and Deep Q-learning are not appropriate choices of learning methods for this adversarial simulation.
- Double Deep Q-learning performs the best, out of the three learning methods compared in this paper, for this adversarial problem.
- A constant average reward trend does not necessarily mean the models do not improve but instead improves at a similar rate.

Acknowledgment

The authors would like to thank our supervisor Erik Berglund for his massive support during all different phases of this project. We would also like to express our gratitude to Ran Li from project group C4b and Johan Wahréus from project group C3b for their feedback on this paper that helped improve both the clarity and the content of the report.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge: MIT Press, 1998, ch. Introduction, pp. 3–23.

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015.

[3] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–489, Jan. 2016.

[4] B. Baker, I. Kanitscheider, T. M. Markov, Y. Wu, G. Powell, B. McGrew, and I. Mordatch, "Emergent tool use from multi-agent autocurricula," *CoRR*, vol. abs/1909.07528, Feb. 2020. [Online]. Available: http://arxiv.org/abs/1909.07528

[5] D. Mwiti. (2021, Feb.) 10 real-life applications of reinforcement learning. Neptune Labs Inc, Warsaw, Poland. [Online]. Available: https://neptune.ai/blog/reinforcement-learning-applications

[6] D. Wang, B. Ding, and D. Feng, "Meta reinforcement learning with generative adversarial reward from expert knowledge," in *2020 IEEE 3rd International Conference on Information Systems and Computer Aided Education (ICISCAE)*, Sep 2020, pp. 1–7.

[7] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits and Systems Magazine*, vol. 9, no. 3, pp. 32–50, Aug 2009.

[8] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge: MIT Press, 1998, ch. The Reinforcement Learning Problem, pp. 51–85.

[9] Abhijit Gosavi. (2019, Sep.) A tutorial for reinforcement learning. [Online]. Available: https://web.mst.edu/~gosavia/tutorial.pdf

[10] S. Jia, L. Shen, and H. Xue, "Continuous-time markov decision process with average reward: Using reinforcement learning method," in *2015 34th Chinese Control Conference (CCC)*, Sep 2015, pp. 3097–3100.

[11] F. Garcia and E. Rachelson, *Markov Decision Processes in Artificial Intelligence*. John Wiley & Sons, Ltd, Feb 2013, ch. Markov Decision Processes, pp. 1–38. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118557426.ch1

[12] T. P. Oikarinen, T. Weng, and L. Daniel, "Robust deep reinforcement learning through adversarial loss," *CoRR*, vol. abs/2008.01976, Aug 2020. [Online]. Available: https://arxiv.org/abs/2008.01976

[13] Y. Zhang, Q. Zhang, and R. Yu, "Markov property of markov chains and its test," in *2010 International Conference on Machine Learning and Cybernetics*, vol. 4, Sep 2010, pp. 1864–1867.

[14] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 4705–4713.

[15] Y. Mohan, S. G. Ponnambalam, and J. I. Inayat-Hussain, "A comparative study of policies in q-learning for foraging tasks," in *2009 World Congress on Nature Biologically Inspired Computing (NaBIC)*, Dec 2009, pp. 134–139.

[16] F. S. Melo. (2021, Mar) Convergence of q-learning: a simple proof. [Online]. Available: http://users.isr.ist.utl.pt/~mtjspaan/readingGroup/ProofQlearning.pdf

[17] B. C. Csáji, "Approximation with artificial neural networks," Master's thesis, Eötvös Loránd University, Budapest, 2001.

[18] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference for Learning Representations (ICLR 2015)*, Jan 2017. [Online]. Available: https://arxiv.org/pdf/1412.6980.pdf

[19] M. Sewak, *Deep Reinforcement Learning Frontiers of Artificial Intelligence*, 1st ed. Singapore: Springer Singapore, 2019.

[20] A. Krogh and J. Hertz, "A simple weight decay can improve generalization," in *Advances in Neural Information Processing Systems*, J. Moody, S. Hanson, and R. P. Lippmann, Eds., vol. 4. Morgan-Kaufmann, Dec 1991, pp. 950–957. [Online]. Available: https://proceedings.neurips.cc/paper/1991/file/8eefcfdf5990e441f0fb6f3fad709e21-Paper.pdf

[21] H. v. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the 13th AAAI Conference on Artificial Intelligence*. AAAI Press, Feb 2016, p. 2094–2100. [Online]. Available: https://dl.acm.org/doi/10.5555/3016100.3016191

[22] T. P. Team. (2021, Apr.) Pypy. A free and open-source software (FOSS) project. [Online]. Available: https://www.pypy.org/

[23] R. Teja. (2021, Apr) Basics of embedded c program. [Online]. Available: https://www.electronicshub.org/basics-of-embedded-c-program/

CONTEXT D
THE CO2-FREE POWER SYSTEM

POPULAR DESCRIPTION

Wind turbines will be swimming with the fishes

The power system is the apparatus that gives our society electricity, and most people can agree that wind power is a central part in the future CO_2-free power system. However, no one seems to want wind turbines in their backyard. Now! There might be a solution to this. New offshore wind farms can barely be seen from shore and floating turbines can be placed even further out.

Floating turbines might sound like a crazy idea, but there is already a demo project in operation. The world's power consumption is constantly increasing, and fossil fueled production of electricity is gradually decreasing. Because of this, new ways to implement renewable energy sources are being brought forth. Floating wind turbines are such an example.

Why would one want to make a floating wind turbine when there already is wind power being generated offshore? The current offshore turbines must stand in shallow waters, due to the fact of a steep increase in costs in deep waters. Floating wind turbines can be placed where the wind conditions are optimal and in countries where the ocean is too deep for conventional turbine foundations.

One offshore wind turbine can produce enough energy to power 6000 homes, which means that only 800 wind turbines would provide enough energy for all households in Sweden. In other words, wind power has great possibilities to eliminate our carbon dioxide emissions. You are probably wondering what happens when it is not blowing enough. Will your household be out of electricity? Absolutely not! With efficient hydropower production planning, sufficient transmission capacity and a sustainable market design we can ensure a stable and CO_2-free energy system

SUMMARY OF PROJECT RESULTS

There is a need to convert more of the world's fossil-based energy production to renewable alternatives in order to minimize the negative effects that carbon-based energy production has on the climate. One suggestion to tackle this problem is to replace the CO_2 generating energy production with renewable alternatives. This poses a challenge on the regulation of energy production since most of the renewable energy sources are not able to be regulated, such as wind power and solar energy. Furthermore, the optimal placements for such energy sources are often found in remote areas, which creates a demand for improved power grids as well as energy distribution centers.

In project D1, a study is done regarding how an increased amount of power generated from wind farms might cause voltage problems within a simulated electrical transmission grid. This covers remote wind farms, as well as large wind parks closer to the power consumers. The project also includes investigating the costs of necessary upgrades in the transmission system, to remedy the negative effects on the voltage with an increased amount of power from renewable sources. To be able to acquire results for the entire simulated transmission grid, the method of load flow analysis is utilized as a numerical method for solving non-linear systems of equations.

For future projects regarding the effects of an increased amount of power from wind and solar, a further investigation in a more realistic and advanced transmission grid could be done. This could include real transmission lines of an entire electricity area, including input and output out of this massive transmission grid. To simulate energy storage in a future transmission grid, an additional study could be done regarding the possibility of including batteries as the method of energy storage in a transmission grid.

In project D2, a possible implementation of a forward capacity market (FCM) in Sweden is examined. This type of market pays producers for their capacity to produce. In traditional energy-only markets, producers earn money by selling electrical energy in MWh. With a large share of weather dependent energy sources with low production cost, the price on electricity is low during a significant part of the year. Consequently, energy sources where the production can be controlled, may experience difficulties covering their annual fixed costs. However, plannable generation capacity is necessary when weather dependent

energy sources provide insufficient power. An FCM could enable plannable power plants to be profitable by paying for the capacity they provide. In the paper, results are presented suggesting that the need for an FCM is limited in Sweden today. However, when nuclear power is phased out, a FCM could become interesting as it ensures enough available capacity.

Future work related to capacity inadequacy includes calculating reliability for the Swedish power grid in order to develop a better model of a capacity market in Sweden. Other ways of keeping plannable capacity profitable could also be an interesting area to investigate, for example improving pricing when electricity is in short supply.

In project D3 the objective is to optimize hydropower plants in a river in order to be able to sustain a challenging load with minimum spillage. When the load is shaped as a square wave, which increases and decreases frequently, it becomes more difficult for the system to maintain a stable output. The group has implemented a model in the modelling software Spine in order to accomplish this. The model contains, real and approximated, data for each power plant and a fictional cost is set for the spillage. From these input parameters the model derives an optimization. The mentioned methods have provided good results when it comes to minimizing spillage while maintaining the load demands. However, the results of this project show that critically low, as well as high, reservoir levels in combination with a quickly changing load results in an increase in spillage.

In future projects for optimization and planning of hydropower plants operation, the theoretical model representing the river system should be more extensive in terms of the included parameters. Based on some observations from the simulations in project D3, it is obvious that the operation patterns of the power plants are unrealistic. In order to have a more reasonable model, it is important to take more aspects into consideration in future works.

The results presented in these projects all support the development of a future CO_2-free power system. Each project in this context faces different challenges, and the transition to a more sustainable society requires deep knowledge in all these areas. Further studies within these contexts could aid that transition.

IMPACT ON SOCIETY AND ENVIRONMENT

The implementation of a completely CO_2-free power generation will have effects both locally and globally. In order to maintain stability in the power system, a further development of plannable power generation such as hydropower, transmission grids, and a well-functioning design of the electricity market is crucial. The projects in this context focus on these three areas and will therefore facilitate a larger share of renewable resources like wind power. The transition to a power system free of carbon dioxide emissions leads to several dilemmas where compromises between social, economic, and environmental sustainability must be made.

An important aspect in the CO_2-free power system is giving proper incentive to build and operate more renewable energy sources. It will be difficult to get companies to invest in such endeavors without the possibility of monetary gain. Action has already been taken in various countries in order to make solar and wind power generation affordable by subventions. According to the UN, access to energy is a human right. This right is more attainable in some countries than in others. In those places where this right is hard to obtain, fossil fuels are often the only applicable solution. This may be because of insufficient power grids or lack of investment capital. Whatever the reason, more resources must be directed into these countries in order to maintain the right to have access to energy while reducing fossil fueled power generation.

Every country has a responsibility to work towards a reduction in CO_2 emissions because the whole globe is affected. A decrease of emissions in one country will not be enough; it must be on a global scale. Those countries that have it easier to convert to CO_2-free power generation should aid the countries where it is more difficult. Furthermore, every able person has a responsibility to help in these efforts. A simple way for citizens to aid this transition is to buy electricity that guarantees the origin, and that this origin should be renewable sources.

Hydropower mainly has an impact on the local ecosystem services. Dams cause interruptions in fish migration routes, which in turn affects their reproductive capacity. Besides that, variations in the flow through a hydropower plant can cause a periodic dry-out of riverbeds which can make it hard for fish to survive. However, power producers are nowadays often obliged to compensate for the negative consequences, for example by planting of fish in nearby lakes and rivers.

Wind power is often regarded as an inexpensive alternative compared to other renewable energy sources. However, when it comes to wind turbines there is no producer responsibility and no requirements for recycling. The aim of putting the responsibility on the producer should be to force wind turbine manufacturers to produce and recycle in a resource-efficient manner with minimal climate impact. The economical sustainability, that comes from implementing wind power in

transmission grids, comes at the cost of efficient usage of resources. It is often seen by manufacturers as too expensive to recycle. Therefore, the impact on the environment from wind power can still be decreased. The ethical dilemma is whether there should be regulations forcing manufacturers to recycle the turbines or not. Our view is that the regulations will lead to lower profitability and may therefore inhibit the expansion of wind power. Although these regulations may temporarily hinder the transition into a CO_2-free power system, it might also accelerate the developments of innovations that lead to a huge leap in the sustainability of a wind turbines life cycle.

The effect on individuals from this transition is clear. The phrase *"not in my backyard"* has recently been a frequently used headline for newspapers in Sweden regarding the implementation of new windmills. It is obvious that the transition will not always be easy. When people build their houses, or expand their existing house, they need a building permit approved by all their neighbors. This is not always the case when constructing wind turbines. Companies in the electric power sector need land to build their wind turbines, and the fastest way can be to purchase rights to build wind farms on someone else's land. Since wind turbines are taller than the average building, they can be seen from a long distance. The result is that a lot of people have to look at large rotating wind turbine blades from their backyards. Not to want wind farms visible from the backyard is therefore a quite reasonable opinion. A lot of people will be forced to have these wind farms in their backyard for the possibility of reaching a completely CO_2-free society. This leads to the question; where do we put all the wind turbines to make that transition a possibility? A large conflict of interest arises between the people that live in our larger cities and the people living more rural as the latter are more affected by the new wind parks.

However, it is important to address the fact that a CO_2-free power system creates enormous opportunities for positive progress towards a more sustainable future. For instance, a major reduction in the levels of CO_2-emissions is a guaranteed outcome when large-scale investments are made in renewable energy sources. According to the Paris Agreement, mankind needs to act towards limiting global warming by reducing greenhouse gas emissions. We must, in other words, start working towards these goals in a way that inspires others to make a change as well. Industries, transports and other large sectors are highly dependent on energy. Offering these sectors the access to green and reliable energy would have a positive impact on both our society and the environment. In other words, a further reduction of CO_2-emissions in these sectors would help us mitigate global climate change and get one step closer to sustainability.

Voltage Deviations in a Power System

Klas Lindgren and Sophia Larbi Engelbrektsson

***Abstract*—The aim of this project was to analyze how the voltage magnitudes of an electrical grid is affected when wind power production varies in an area around the river of Ångermanälven. The goal of the project was to keep the voltage deviation within 10 % from the set base value. A secondary goal was to make a profitability assessment between power losses and costs related to the power grid.**

A transmission grid model was built around Ångermanälven and simulations were made in MATLAB, with the open-source tool package called MATPOWER, to simulate the properties of the grid. These simulations included real hourly historical data for demand and power generation. Voltage deviation and losses in the transmission grid for the system was then determined with power flow analysis.

For the base case, the voltage deviation was kept within the limit of a maximum deviation of 10 %. The base case was thereafter upgraded to improve transmission efficiency and resiliency. Increasing the base voltage resulted in lower losses and voltage deviations below 5 %. To make the grid more resilient and fulfill the N-1 and N-2 criteria, additional transmission lines were added. However, these were deemed necessary for a reliable grid, even though the upgrades increased the total cost of the system.

***Sammanfattning*—Strävan med detta projekt var att analysera hur spänningsnivån för ett elnät påverkas av varierande vindkraftsproduktion för ett område kring Ångermanälven. Målet med projektet var hålla spänningen inom 10 % från den satta basspänningen. Ett sekundärt mål var att göra en lönsamhetsbedömning mellan effektförluster och kostnader relaterade till elnätet.**

En model av transmission nätet byggdes kring Ångermanälven och simuleringar utfördes i MATLAB, med hjälp av ett open-source verktyg kallat MATPOWER, för att simulera nätets egenskaper. Simuleringarna inkluderade verklig historisk timvis data för behov och kraftproduktion. Spänningsavvikelser och förluster i transmissionsnätet fastställdes med belastningsfördelning.

För basfallet hölls spänningsavvikelserna inom den maximala gränsen på 10 %. Basfallet uppgraderas därefter för att förbättra transmissionseffektivitet och tillförlitlighet. Ökning av basspänningen resulterade i lägre andel förluster och spänningsavvikelser på under 5 %. För ett mer tillförlitligt nät och för att kunna uppfylla N-1 och N-2 kriterierna, installerades extra ledningar. Dessa ledningar ansågs nödvändiga för att uppnå ett tillförlitligt elnät, även om det innebar ett krav på ökade investeringskostnader för systemet.

***Index Terms*—MATPOWER, Voltage Deviations, Power Flow, Power System, Wind Power, Resiliency**

Supervisors: Lennart Söder and Evelin Blom

TRITA number: TRITA-EECS-EX-2021:154

I. Introduction

A. Background

Since the first electric light bulb was lit, the generation of electricity has taken place in large, centralized power plants. As society now moves forward towards a CO_2-free power system, the amount of electricity being produced in smaller plants are increasing [1]. These changes are affecting the market in such a way that the expensive power plants, for instance nuclear power plants, are going out of fashion. The low operating costs of wind turbines, and favorable wind speeds, periodically drives down the prices of electricity to the point that nuclear power plant owners can't make money due to their higher production costs [2].

The continued increase of electricity from wind power turbines and the shutdown of nuclear power reactors have led to the power system facing new challenges. Electricity from modern power plants, such as wind and solar farms, are immensely variable, and take place when the weather conditions allow for power generation. Therefore, electricity production takes place to a greater extent when weather conditions are optimal for electricity production, but not necessarily when the demand for electricity is high.

To be able to meet demand it is required that the electricity system have enough capacity for power transmission and generation during peak demand. In the modern-day transmission grid, limits in the power transmission capacity causes stress in the electricity system. Power shortages and stress occurs when the demand for electricity is higher than what is physically possible to be delivered. When even more conventional power plants are shut down and weather-based power production increases, stress in the electricity distribution system and its design will increase [3].

With the increasing electricity demand in our, ever increasing, electrified society, unreliable power production can also lead to costly power outages. When converting to a power grid containing more unreliable weather-dependent power production, an increased amount of flexible power capacity is required in the system to aid during power shortages [4]. Flexible power, such as hydro power, thus becomes a necessity in systems that consist of a high proportion of wind power.

When the demand for electricity cannot be met, during a specific time, due to limits in the production and transmission capacity, voltage deviations in the system may be affected.

When large power consumers, such as Stockholm, have exceptionally high demand, the transmission grid is working close to its limits. During these periods there are

huge power flows in the transmission lines from the hydro power plants and wind farms in the north. If an accident occurs and causes a fault in the transmission grid, immense repercussions on the availability of electricity can occur.

If not the entire existing, already strained, transmission grid can be used to supply all the consumers with electricity, the voltages in the system might be affected.

Faults in the electric power transmission grid are often caused by external factors on the electric power transmission wires and cables. Faults on the overhead transmission are usually rectified within 24 hours, while the time to repair faults in the underground transmission varies from a week up to a month [5]. To avoid costs connected to faults and power losses in the transmission grid, a more well-developed transmission grid will be needed.

In this project, wind farms will complement the existing hydro power for the electric power generation needed in the grid to meet demand. It is important to meet the demand this since societal costs are very high if production cannot meet the current demand. Furthermore, an investigation will be made how the voltage in the system is affected by varying wind power production and demand.

An additional study will be made to investigate how to design a resilient, and fail-safe, electric power transmission grid, even when faults occur on individual wires and cables. A fail-safe system is a system capable of continue working within the specified limits, even when a certain failure arises.

Trade-offs will also be made about which components and voltage levels are most profitable for the system.

B. Goals

The main goal of this project is to achieve a power grid system with varying production from wind farms with a maximum of 10 % voltage magnitude deviation from a selected reference value. To achieve this, a designed model of a power grid, surrounding the river of Ångermanälven, will be utilized. The goal of the project is not to simulate the real grid, but to try to analyze how the transition to more wind power might affect voltage levels in a future transmission grid.

The created model will then be used to simulate how the grid behaves, with the help of power flow analysis. For the construction of the electric power transmission grid, interconnected power lines, *branches*, will be put between nodes, *buses*, in a way that would be similar of the real transmission grid. The grid will be designed to be resilient and work according to the N-1 and N-2 criteria to ensure power supply, even if faults such as a transmission line failure should occur. Power flow within the system will be simulated using hydro power, wind power and demand from historical data.

The secondary goal of this paper is to make a profitability assessment. This assessment aims to investigate the relationship between losses and costs when choosing suitable components for the system. To accomplish this, different components, power lines, and wind turbines, will be analyzed in the simulations to see which achieves minimal power loss in the system, in comparison to the size of the investments. The goals for this project is therefore to:

- Build a working model of a transmission grid around Ångermanälven;
- With voltage deviations not exceeding 10 % from a selected base voltage; but also to
- Design a resilient grid, that fulfills the N-1 and N-2 criteria; and finally
- Make a profitability assessment of improvements made to the power system.

II. Theory

To understand the calculations being done in the simulations, some theoretical background is required. The theory behind all analysis and simulation throughout this project is explained in this section. As the simulations are made with MATPOWER [6], a tool package for MATLAB [7], the theory behind the calculations made in MATLAB will also be explained.

As an introduction to all calculations, the theory behind how three-phased power lines can be analyzed will be dealt with. The section will cover the components that can theoretically be used to model a power line when modeling a transmission grid. The concepts regarding buses, line models for transmission lines, admittance matrices, power losses in the transmission system and power flow of systems will also be covered. The section about admittance matrices explains in depth about how the model can be implemented to calculate the voltage deviations in the system.

As a mathematical background, Newton's method will be covered as an introduction to the theory on power flow. Newton's method is a method to numerically obtain approximate roots of functions.

Finally, the power flow section then ties everything together and explains how to simulate power flow in a desired system. These sections will stand as a foundation for the section covering what MATPOWER utilizes in its calculations.

A. Power Lines

Power lines, whether overhead wires or underground cables, consist of a *resistance, (r)*, *shunt conductance, (g)*, *inductance, (l)* and *shunt capacitance, (c)*.

The resistance occurs due to conductor resistivity, while current leakage in isolation causes shunt conductance. Likewise, inductance is caused by magnetic flux whereas shunt capacitance is caused by the electric field between the ground and the lines, as well as in between the lines.

Assuming the lines can be approximated as a symmetrical three phase, a model of the line can then be used to visualize the quantities of these values [8]. This model is shown in Fig. 1.

To calculate the resistance in power lines, equation (1) is used.

$$r = \rho \frac{L}{A} \quad [\Omega] \tag{1}$$

Where ρ is the material property of the material used in the wire called resistivity, ρ. Resistivity is given in ohmmeters (Ωm). L is the length (m) of the wire. A is the cross sectional area (m^2) of the wire [9].

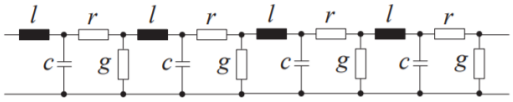


Fig. 1. Symmetrical three-phase power line model [8].

Shunt conductance is frequently disregarded in power line calculations. The data for shunt conductance vary from case to case depending on the air humidity as well as air pollution [8]. Therefore, shunt conductance was disregarded in this paper.

Inductance is related to the reactance, *(X)*, in equation (2), which is the imaginary part of impedance *(Z)*. The inductance of the power line plays a vital role in voltage fluctuations and line losses, while the reactance affects both voltage fluctuations and reactive powers in the system. How the reactance is included in the system is shown in Fig. 6. Inductance in a power line can be calculated by using equation (3).

$$X = 2\pi f l \quad [\Omega/km, phase] \tag{2}$$

$$l = 2 * 10^{-4}(\frac{1}{4n} + \ln(\frac{a}{(\frac{d}{2})})) \quad [H/km, phase] \tag{3}$$

Where n represents the number of conductors in each phase. While d (m) is the conductors diameter, a (m) is the geometrical mean distance, as shown in Fig. 2.

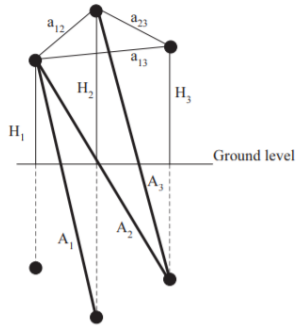


Fig. 2. Geometrical quantities for calculation of inductance and capacitance for a power line [8].

To calculate the mean distance a, equation (4) is used.

$$a = \sqrt[3]{a_{12}a_{13}a_{23}} \quad [m] \tag{4}$$

To acquire a valid result for the inductance, a few assumptions and circumstances is needed to make the calculations a possibility. The wire in a three-phase transmission line needs to be *transposed*, which most long overhead wires are. The phenomenon of transposing a three-phase overhead wire is illustrated in Fig. 3.

Implementation of transposing cycles results in all conductors having the same distance to the ground in each phase as well as equal distance to each other. Furthermore, by having the same distance for all conductors in each phase the inductance will be equal in all three-phases [8].

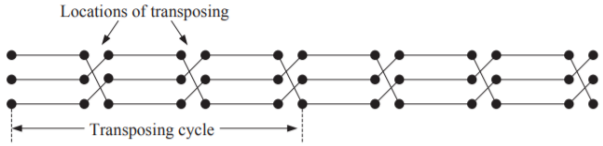


Fig. 3. Transposing of a three-phase power line [8].

To calculate the shunt capacitance for each phase of a transposed power line equation (5a) is used.

$$c = \frac{10^{-6}}{18\ln(\frac{2H}{A}\frac{a}{(d/2)_{eq}})} \quad [F/km, phase] \tag{5a}$$

$$(d/2)_{eq} = d/2 \tag{5b}$$

$$(d/2)_{eq} = \sqrt[n]{n(D/2)^{n-1} * (d/2)} \tag{5c}$$

Where $(d/2)_{eq}$ is an expression for the diameter of the conductors used in each phase. If only one conductor per phase is used, equation (5b) is needed. If there are multiple conductors used in each phase, equation (5c) is the appropriate equation to be used when solving the capacitance with equation (5a). Also, n is the number of conductors per phase while D is the diameter of the circle formed by the multiple conductors shown in Fig. 4.

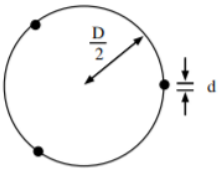


Fig. 4. Example of three conductors used per phase [10].

The distances A and H are calculated by using the geometrical mean value. H is a mean height, from the ground to the overhead line, that can be calculated with equation (6). A is a theoretical mean distance and is calculated with equation (7). The distances are illustrated in Fig. 2. The value of A_i is the distance between the conductors and their mirror image. H_i is the height between conductors and ground level [10].

$$H = \sqrt[3]{H_1H_2H_3} \quad [m] \tag{6}$$

$$A = \sqrt[3]{A_1A_2A_3} \quad [m] \tag{7}$$

The shunt susceptance (b) may also be included when modeling a three-phase transmission line. Equation (8) is used to calculate the susceptance *(Siemens)* in each phase.

$$b = 2\pi f c \quad [S/km, phase] \tag{8}$$

Where f is the frequency, which has a nominal value of 50 Hz in Sweden, and c is the shunt capacitance calculated with equation (5a) [8].

B. Load flow study

When analyzing power systems, three different types of buses are used for power flow analysis. These buses are called *PQ-bus, PU-bus* and *Slack bus*. The known data and input for each bus decides if a bus is assigned as a PQ, PU, or a slack bus [10].

PQ-bus: In a PQ-bus the active power, P, and the reactive power, Q, is specified. A PQ-bus is also known as a *load* bus because the demanded load is specified. In this node the voltage, V, and its phase angle Θ is unknown [10].

PU-bus: In a PU-bus the active power, P, and the magnitude value of the voltage, U, is specified. Meanwhile, the reactive power, Q, and the phase angle Θ is unknown [10]. The PU-bus is often called a *generator* bus.

Slack bus: The slack bus can be seen as a bus that contains large generator that can freely adjust its real and reactive power output, so that the power flow can be solved. [10]. It can also be assumed that the slack bus contains a variable load, to be able to adjust its power input. The voltage magnitude in the slack bus is always known, 1 p.u., while the phase angle is 0. The purpose of the slack bus is therefore to inject or withdraw power from the system to make the system of equations solvable [10].

Per Unit (p.u.): Per unit system is a common method to express voltage, power, current and impedance in electrical systems. Per unit denotes a value in fraction compared to a reference value. For instance, a voltage base or power base can be selected as references to calculate the voltages and power flows in the entire system [10]. One of many, advantages of using per unit, is that is eases the computation. The per unit method simplifies the power flow calculations, as all the quantities are only expressed in term of p.u. The different base values used in this project is a base for the apparent power, S, and the voltage, U. With these two base values, a base for the impedance, Z, and the current, I, can then be calculated.

$$p.u. = \frac{real\ value}{base\ value} \tag{9}$$

Power Factor: The power factor, φ, is the phase shift between the voltage and the current. In equation (10) the relationship between active and reactive power in relation to the power factor is explained. The X represent a reactance that could be either inductive or capacative.

$$tan(\varphi) = \frac{X}{R} = \frac{Q}{P} \tag{10}$$

$$S = \sqrt{P^2 + Q^2} \tag{11}$$

Apparent power can be shown in the complex plane as a power triangle as in Fig. 5. Where *Imaginär axel* denotes the imaginary axis, while *Reell axel* denotes the real axis and φ denotes the power factor [11].

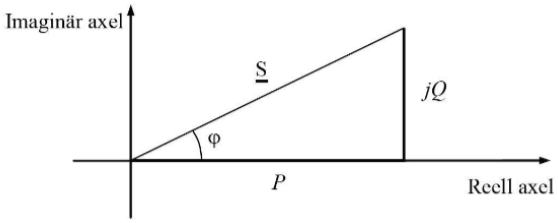


Fig. 5. Apparent power in the complex plane [11].

C. Line model

To simulate a transmission line, a π-transmission line model can be implemented. The equivalent and nominal π-circuit for a three-phase transmission line consist of an impedance in series with the transmission line for each phase. The shunt effect on the transmission line is represented by splitting the shunt admittance in two and representing them as two capacitors. One at the sending end of the line and the other at the receiving end [12]. The capacitors are then placed between the transmission line and the ground. The model of the π-equivalent line is illustrated in Fig. 6. Where Z is the line impedance, Y is the shunt admittance, U and I represents voltages and currents, respectively.

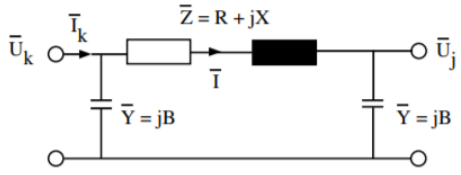


Fig. 6. Equivalent π model of a transmission line [10].

The π-equivalent line model can be utilized to model transmission lines. For shorter lines susceptance can be assumed to not affect the results. The shunt admittance Y is calculated in equation (12). Where b is the shunt susceptance (S) of the capacitor and s *(m)* is the length of the line [10]. The imaginary unit is represented by j.

$$j = \sqrt{-1}, \quad j^2 = -1$$

$$\bar{Y} = jB = j\frac{bs}{2} \quad [S/m] \tag{12}$$

Line losses can be calculated by utilizing the π-transmission line model. The properties of the model is a good approximation of a real transmission line. By using the variables in Fig. 6, voltages at different buses can be calculated using equations (13a) with (13b), to get the voltage with equation (13c):

$$\bar{U}_j = \bar{U}_k - \sqrt{3\bar{Z}\bar{I}} \tag{13a}$$

$$\bar{I} = \bar{I}_k - \bar{Y}\frac{\bar{U}_k}{\sqrt{3}} \tag{13b}$$

$$\Rightarrow \bar{U}_j = (1 + \bar{Z}\bar{Y})\bar{U}_k - \sqrt{3\bar{Z}\bar{I}_k} \tag{13c}$$

D. Admittance matrices

To calculate the voltage deviation in the system using equation (13b) or (13c), the *admittance matrices* of the

system is needed. In Fig. 7 an example with four buses is illustrated.

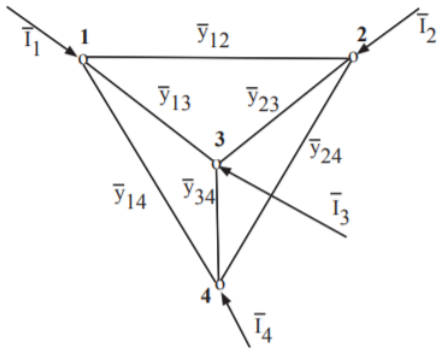


Fig. 7. A four bus system with admittance [10].

Buses, *nodes*, are tied together in a network with branches, *transmission lines*, and these branches have an admittance $\bar{y}_{kj}$. Where kj refers to the index of the buses the admittance is associated with. The voltage in each bus within the system is described as $\bar{U}_1, \bar{U}_2, \bar{U}_3$ and $\bar{U}_4$, while $\bar{I}_1, \bar{I}_2, \bar{I}_3$ and $\bar{I}_4$ are currents injected in each bus from external current sources.

With Kirchhoff's current law, *KCL*, the balance equations can be obtained for the system of buses. For bus 1, in Fig. 7, the balance equation are presented in equation (14) and (15), where the final result in equation (15) uses the set of substitutes presented in equation (16). Where $\bar{Y}_{kk}$ is the sum of all admittances connected to *node k*, and $\bar{Y}_{kj}$ is the admittance between *node k* and *node j*.

$$\bar{I}_1 = \bar{y}_{12}(\bar{U}_1 - \bar{U}_2) + \bar{y}_{13}(\bar{U}_1 - \bar{U}_3) + \bar{y}_{14}(\bar{U}_1 - \bar{U}_4) \tag{14}$$

$$\begin{aligned}\bar{I}_1 &= (\bar{y}_{12} + \bar{y}_{13} + \bar{y}_{14})\bar{U}_1 - \bar{y}_{12}\bar{U}_2 - \bar{y}_{13}\bar{U}_3 - \bar{y}_{14}\bar{U}_4 \\ &= \bar{Y}_{11}\bar{U}_1 + \bar{Y}_{12}\bar{U}_2 + \bar{Y}_{13} + \bar{Y}_{14}\bar{U}_4\end{aligned} \tag{15}$$

$$\begin{aligned}\bar{Y}_{11} &= \bar{y}_{12} + \bar{y}_{13} + \bar{y}_{14} \\ \bar{Y}_{12} &= -\bar{y}_{12} \\ \bar{Y}_{13} &= -\bar{y}_{13} \\ \bar{Y}_{14} &= -\bar{y}_{14}\end{aligned} \tag{16}$$

Constructing balance equations, with the same procedures as above, for bus 2,3 and 4, an admittance matrix can be created [8]. See equation (17).

$$\mathbf{I} = \begin{bmatrix}\bar{I}_1 \\ \bar{I}_2 \\ \bar{I}_3 \\ \bar{I}_4\end{bmatrix} = \begin{bmatrix}\bar{Y}_{11} & \bar{Y}_{12} & \bar{Y}_{13} & \bar{Y}_{14} \\ \bar{Y}_{21} & \bar{Y}_{22} & \bar{Y}_{23} & \bar{Y}_{24} \\ \bar{Y}_{31} & \bar{Y}_{32} & \bar{Y}_{33} & \bar{Y}_{34} \\ \bar{Y}_{41} & \bar{Y}_{42} & \bar{Y}_{43} & \bar{Y}_{44}\end{bmatrix} \begin{bmatrix}\bar{U}_1 \\ \bar{U}_2 \\ \bar{U}_3 \\ \bar{U}_4\end{bmatrix} = \mathbf{YU} \tag{17}$$

The more general case, with n number of buses, the admittance matrix would be:

$$\mathbf{I} = \begin{bmatrix}\bar{I}_1 \\ \vdots \\ \bar{I}_n\end{bmatrix} = \begin{bmatrix}\bar{Y}_{11} & \cdots & \bar{Y}_{1n} \\ \vdots & \ddots & \vdots \\ \bar{Y}_{n1} & \cdots & \bar{Y}_{nn}\end{bmatrix} \begin{bmatrix}\bar{U}_1 \\ \vdots \\ \bar{U}_n\end{bmatrix} = \mathbf{YU} \tag{18}$$

The admittance matrices can then be used in solving power flow problems. More of this in section *II-F. Power flow of systems.*

E. Power losses in the transmission system

Active power losses, P_L, in a transmission line is due to the real part of impedance: the resistance R, and magnitude of the line current, I. The loss of active power in a three-phase power line is presented in equation (19).

$$P_L = 3RI^2 \tag{19}$$

The line current, I, can be rewritten like equation (20). Where S is the apparent power, P active power and Q the reactive power injected in the system.

$$I^2 = \frac{S^2}{3U^2} = \{S = \sqrt{P^2 + Q^2}\} = \frac{P^2 + Q^2}{3U^2} \tag{20}$$

The active power loss would then be:

$$P_L = R_{kj}\frac{P_{kj}^2 + (Q_{kj} + BU_k^2)^2}{U_k^2} \tag{21}$$

In equation (21), the kj indexes references the number of the bus. The susceptance, B, multiplied by the square of the voltage, U_k, is the reactive power produced by the existing shunt capacitance at bus k. Additionally, equation (21) shows that an increase of voltage would decrease the active power losses in the system. While an increase of transmitted active power would increase the active power losses [8].

Similarly, to the active power losses, there are also reactive power losses, Q_L, in the system. The reactive power losses are due to the imaginary part of the impedance, X, called *reactance*. The reactance is displayed in Fig. 6 as a part of the impedance, Z. Equation (22) is very similar to equation (21), except to the imaginary part X, compared to the real part R.

$$\Re e\{Z\} = R, \quad \Im m\{Z\} = X$$

$$Q_L = X_{kj}\frac{P_{kj}^2 + (Q_{kj} + BU_k^2)^2}{U_k^2} \tag{22}$$

Reactive losses in the line results in an increase of the total losses for the transmission line [8].

F. Power flow of systems

Newton's Method: The results of the power flow is obtained by iterative calculations, and an explanation of Newton's method is therefore seen as a necessary introduction.

Newton's method, also known as Newton-Raphson method, is a tool for numerical analysis and is used find approximate roots for equations and functions.

$$f(\mathbf{x}) = \mathbf{0} \tag{23}$$

Newton's method for single variable functions is usually displayed as in equation (24), where k is the iteration count [13].

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, ... \tag{24}$$

For the first iteration, x_0 is an approximated value, guessed as a root (a solution) for the function $f(x)$.

If the function f is continuously differentiable and the approximated first guess is close enough to the correct root, the successive approximations will converge to a root. Newton's method has a quadratic rate of convergence and can be used to find multiple solutions to functions containing mulitple variables.

If the function f is a vector containing a system of multivariable functions it is presented as in equation (25).

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{J}^{-1} f(\mathbf{x}_k), \quad k = 0, 1, ... \tag{25}$$

Where $\mathbf{J}$ is the Jacobian matrix containing the first-order partial derivatives of the functions with regard to all the variables, $J_{ij} = \frac{\partial f_i}{\partial x_j}$.

$$\mathbf{J} = \frac{d}{d\mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{x_n} \end{bmatrix} \tag{26}$$

When the desired accuracy of the solution is achieved, that is when the approximated solutions is sufficiently close to the actual solution as specified by the error tolerance, the iterative process can be stopped [13].

Load flow: Power flow analysis, is a method used to determine the power properties of a bus in a network of buses and branches. When voltages, and both their magnitudes and angles, are known for buses in a system, other properties, such as line losses, can be calculated. To be able to use load flow analysis, the voltage in the system needs to be determined.

Applying the Newton's method for a load flow problem, the partial derivatives are used to find the difference in the independent variables. Which is then used to find a solution for the system and its properties. All values used in the equations for Newton's method will be expressed in per unit [8].

The load flow will be done in snapshots, one snapshot every hour, to recreate the system conditions for a specific point in time. These conditions is what is solved iteratively.

Step 1: The admittance matrices needs to be created, as stated in equation (17), and the net active and reactive power production needs to be calculated, by using equation (27) and (28). For the first iteration, initial per unit values needs to be approximated for the voltages, U, as well as the phase angles Θ [8].

$$P_{GDk} = P_{Gk} - P_{Dk} \tag{27}$$

$$Q_{GDk} = Q_{Gk} - Q_{Dk} \tag{28}$$

Step 2: The injected power into each bus needs to be calculated, and then used to determine the difference between the net production from equation (27) and (28) and the injected power in each bus. This is to identify any power surplus or shortage, that needs to be adjusted for with the slack bus.

The simulated system uses π-transmission line models, as the one presented in Fig. 6. The apparent power, S, can be divided into active power, P, and reactive power, Q, with the usual methods explained earlier in this paper.

The injected power in bus k is calculated according to equation (29) and (30) respectively [8].

$$P_{kj} = \frac{R_{kj}}{{Z_{kj}}^2} {U_k}^2 + \frac{U_k U_j}{{Z_{kj}}^2}(X_{kj} sin\theta_{kj} - R_{kj} cos\theta_{kj}) \tag{29}$$

$$Q_{kj} = -B{U_k}^2 + \frac{X_{kj}}{{Z_{kj}}^2} {U_k}^2 - \frac{U_k U_j}{{Z_{kj}}^2}(R_{kj} sin\theta_{kj} + X_{kj} cos\theta_{kj}) \tag{30}$$

In Fig. 8, a bus k with connections to N buses is illustrated. Where $\bar{I}_{Gk}$ is the current generated from a generator in bus k, in per-unit. Whereas $\bar{I}_{LDk}$ is the current that the load at the bus demands, also in per-unit. Finally $\bar{I}_{k1}$,$\bar{I}_{k2}$ and $\bar{I}_{kN}$ represents the current going from bus k to bus 1,2 and N.

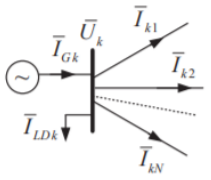


Fig. 8. Bus k in a system (including notations) [8].

Kirchhoff's current law, KCL, then states that the difference between the generated current and the current demand must be equal to the sums of currents from bus k, $\bar{I}_{kj}$, to neighboring buses. This is because the sum of all currents injected at bus k must be zero. See equation (31).

$$\bar{I}_{Gk} - \bar{I}_{LDk} = \sum_{j=1}^{N} \bar{I}_{kj} \tag{31}$$

By multiplying the bus voltage with the conjugate of the current presented in equation (31), equation (32) can be obtained.

$$\bar{U}_k (\bar{I}_{Gk} - \bar{I}_{LDk})^* = \bar{U}_k \bar{I}^*_{Gk} - \bar{U}_k \bar{I}^*_{LDk} = \sum_{j=1}^{N} \bar{U}_k \bar{I}^*_{kj} \tag{32}$$

Furthermore, equation (32) can be expressed, in per unit, as complex power, see equation (33).

$$\bar{S}_{Gk} - \bar{S}_{LDk} = \sum_{j=1}^{N} \bar{S}_{kj} \tag{33}$$

Therefore, the sum of all injected power is given by equation (34) and (34). This is done by separating the complex power, S, into active power, P, and reactive power, Q.

$$P_k = \sum_{j=1}^{N} P_{kj} \tag{34}$$

$$Q_k = \sum_{j=1}^{N} Q_{kj} \tag{35}$$

Using the above equations for all n buses, gives the equations for the active power. As represented in equation (36).

$$\begin{aligned} P_1 &= U_1(Y_{11}U_1 + \ldots + Y_{1n}U_n) \\ &\vdots \\ P_n &= U_n(Y_{n1}U_1 + \ldots + Y_{nn}U_n) \end{aligned} \tag{36}$$

The difference between the net power production and injection is then given by equation (37) and (38).

$$\Delta P_k = P_{GDk} - P_k \tag{37}$$

$$\Delta Q_k = Q_{GDk} - Q_k \tag{38}$$

Step 3: For every iteration small changes are made to the buses in the voltage vector ($\Delta\mathbf{U}$), which causes minor active power deviations, in the active power vector ($\Delta\mathbf{P}$). With these small changes in $\Delta\mathbf{U}$ and $\Delta\mathbf{P}$ a linearized approximation can be obtained for the non-linear system. The approximation then estimates the active power change due to the voltage deviations.

$$\Delta\mathbf{P} = \begin{bmatrix} \Delta P_1 \\ \vdots \\ \Delta P_n \end{bmatrix} = \begin{bmatrix} \frac{\partial P_1}{\partial U_1} & \cdots & \frac{\partial P_1}{\partial U_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial P_n}{\partial U_1} & \cdots & \frac{\partial P_n}{U_n} \end{bmatrix} \begin{bmatrix} \Delta U_1 \\ \vdots \\ \Delta U_n \end{bmatrix} \tag{39}$$

$$\Delta\mathbf{P} = \mathbf{J}\ \Delta\mathbf{U}$$

$$\Delta\mathbf{U} = \mathbf{J}^{-1}\Delta\mathbf{P} \tag{40}$$

Where $\mathbf{J}$ is the *Jacobian matrix*. In the Jacobian matrix every element is the partial derivative of the active power with respect to the voltage. A new Jacobian matrix must be computed when one or more voltages change significantly.

Step 4: When solving the power flow with alternating current, the matrix for active and reactive power can be arranged as in equation (41). Where $|U|\angle\theta$ denotes the complex bus voltage written in polar form [14] and $\mathbf{J}$ is the Jacobian matrix.

$$\begin{bmatrix} \Delta\theta \\ \Delta|U| \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} \tag{41}$$

The estimated voltage after one iteration is then determined with equation (42). For every iteration $\mathbf{U}_{old}$ is replaced by the previous $\mathbf{U}_{new}$, and this is repeated until the value of the power deviation is small and the solution fulfills the decided accuracy of the calculation. Similarly, θ_{new} can be calculated.

$$\mathbf{U}_{new} = \mathbf{U}_{old} - \mathbf{J}^{-1}\Delta\mathbf{P} \tag{42}$$

Step 5: When the numerically analyzed solution fulfills the decided accuracy of the calculation, the iterative process can stop.

The numerically approximated values for the voltage magnitudes and phase angles can then be used to calculate P_k, Q_k, P_{kj} and Q_{kj}, and the load flow is complete. Results such as generated powers, power flows, losses, voltage magnitudes and voltage phase angles can be calculated from the solutions.

Active power losses are given by equation (43), while reactive power losses are stated in equation (44). These are determined by taking the difference between injected power of two connected buses. Resulting in that the losses must be the difference between these [8].

$$\Delta P_{Lkj} = P_{kj} + P_{jk} \tag{43}$$

$$\Delta Q_{Lkj} = Q_{kj} + Q_{jk} \tag{44}$$

The generated power in each bus is given by equation (45) and (46), where P_k denoted the inject power in bus k and P_{GDk} is the net active power production in bus k.

$$P_k - P_{GDk} = 0 \tag{45}$$

$$Q_k - Q_{GDk} = 0 \tag{46}$$

G. Resilient transmission grids

To create resiliency in an electrical grid, the grid must be able to withstand failures without breaking down and violate limitations. The *N-1 criterion* means that a failure of one component should not affect the safety limitations of the power system. Safety limitations in the transmission grid means to keep the voltage and nominal frequency within acceptable limits, as well as being able to meet the demand for all consumers connected to the grid. Svenska Kraftnät, the Swedish power authority in charge of grid reliability, utilizes this criterion for almost all parts of the transmission grid. The name comes from that the power system had N components before the failure and N-1 after a single failure. Additionally, Svenska Kraftnät also applies the N-2 criterion, however it is only applied for critical parts of the system, such as big cities [15]. The N-2 criterion works in the same way as the N-1 criterion, but requires that no violation of limits are occurring, even when there are a two component failure.

For this project, the safety limitation is the voltage deviation. In a resilient simulated grid no voltage deviation should exceed 10 % of the set per unit base value.

III. MATPOWER

MATPOWER is an open-source tool package for MATLAB. The package consists of MATLAB scripts and functions (.m-files) that assists with solving different power flow simulation problems. MATPOWER's intended area of application is to be used as a tool to assist with simulations for researchers in the electrical and electronic engineering field [16].

For power flow simulations assisted by MATPOWER, Newton's method is applied. For every iteration, the polar form and the Jacobian are updated [16], as presented in equation (41).

The scripts utilizes standard steady-state models to analyze power flows of a designed system. Steady state is when the frequency is assumed to be constant, as well as when there are no transient changes in the power flow or voltage.

To aid with the calculations presented in this paper, MATLAB and MATPOWER have been used. Due to the

strengths of the programming language in handling large amount of data in matrices and vectors, all programming and coding is done in MATLAB.

With MATPOWER, it is possible to define and return a single MATLAB structure array that enabled implementation of all the inputs for the system, but also to extract results. In this structure array there are different fields containing different data. For this study, four out of five fields have been used. These are *baseMVA, bus, branch* and *gen*, with the optional unused field being *gencost* [16]. MATPOWER also uses the standard π*-model* for transmission lines with the traditional series impedance and shunt susceptance [16].

Firstly, to use MATPOWER as a tool to assist with *power flow simulations*, a *case file* is needed. In this *.m-file* the entire system, except the *baseMVA*, was defined. The system was defined with *bus data*, *generator data* and *branch data*, and together they create a functional simulated transmission grid.

Secondly, to use the case file containing the designed system and calling the MATPOWER functions [16], a different script was needed to be made. The second file contains all the actual programming by the user, including the yet missing part of the structure array, *baseMVA*. To aid the ease of access and the overall understanding of the inputs made by the user, most of the data used in the case file is specified in the second script, the *main file*, and then transferred to the case file.

A. Case file

In MATLAB the user can assign each bus as one of the buses mentioned in section *II-B. Load flow study.* The hydro power plants were set to PU-buses, while the cities and the wind farms were set to PQ-buses. To be able to solve the power flow, one additional bus was included, the slack bus. In bus data, both active and reactive power demand is set. All included data is loaded into the MATPOWER case structure array called *mpc* [16].

Exactly what input data is used for the *mpc* is found in *Appendix A. MATPOWER structure array.*

To include generators in the simulated transmission grid, the available generators needs to be included and activated in the generator data. The buses with included generators, that is hydro power plants and wind farms, were set as generators. The active and reactive power generated in each bus was set as an input here.

As for the final part of the case file, branch data is needed. The transmission lines and how they connect the buses are stated, as well as the lengths and characteristics of these. The value for the characteristics of the line is calculated by the user, and then included into the model in used for simulations with MATPOWER. Consequently, it is possible to simulate power lines with different properties by adjusting the input.

B. Main file

Firstly, general data is entered. The *baseMVA* is set as the base for the apparent power, S, along with other base values. These base values are used as references for the, *per unit*, power flow calculations.

Secondly, all data used in the simulations are imported. These values are imported from a *.xlsx (Microsoft Excel-file)*, and converted into matrices. The data imported are hourly values for power demand; power generation from both wind and hydro; datetime vectors; and more.

After importing all data, data processing was needed. Datetime vectors of the entire year of 2019 was converted into weeks with their associated week numbers. Generator and consumer data related to the corresponding buses.

Before starting the hourly snapshot simulations the system needed to be loaded, and the settings and constants needed to be defined. One important setting was to enforce the reactive power limits that was set by the user. MATPOWER works in a way so that if any generator has a violated reactive power limit, its reactive injection is fixed at the limit, the corresponding bus is then converted to a PQ-bus and the power flow is solved again. This procedure will be repeated, until there are no more violations [16].

During the study, two different time frames were used. When wanting to look at very specific scenarios, like the *N-2 criterion*, only a few hours were of interest. When looking at the main subjects of the study, all four weeks were desirable.

To analyze the different time-periods, two *for-loops* were used in MATLAB [7]. The first loop is to group the results in different arrays depending on the week. The second loop is set to make one snapshot power flow simulation every hour of the week set in the first loop.

These two loops together, resulted in four laps of the outer loop, while each inner loop consisting of one lap for every hour of every week. The result of this is 4 times 168 loops, which is equal to 672 outputs that were needed to be processed.

Results could then be extracted for processing, by storing them into vectors, matrices, or arrays. When all data is extracted and processed, results could be presented in graphics, such as plots and tables.

IV. Case Study

To create a system surrounding the river of Ångermanälven, a total of sixteen buses needed to be decided for. To represent a somewhat real scenario, the included cities and hydro power plants are existing real ones.

In Fig. 9, a visual presentation of the created system for analysis and simulation is displayed. This includes six cities, six hydro power plants, three wind farms, the transmission lines, and a slack bus (the blue marker).

A. Power consumers (i.e. cities)

To include loads in the system, numerous cities with their hourly demands were included. Sundsvall, Sollefteå, Strömsund, Östersund, Vilhelmina and Örnsköldsvik were chosen as the cities to be included. These cities were chosen because they were the biggest consumers in the area

Fig. 9. An overview of the created system used in simulations [17].

surrounding the river of Ångermanälven. Furthermore, the cities should be both in the eastern area as well as the western area to make the simulated system more realistic. The cities and its bus numbers are displayed in Table IV.

These cities and their demand were retrieved from Statistikmyndigheten, SCB [18], for the years 2009-2018. The historical data for hourly demand in SE2 used was from the year of 2019, and obtained from Svenska kraftnät [19]. Sweden is divided into four different electricity trading areas. From SE1 in the north to SE4 in the south [20]. The system lies within SE2. The available transmission capacity may vary and congest the flow of power between these areas, and therefore different area prices are established [21].

Data for yearly power demand did not exist for all cities for every year, so an annual average demand was calculated for each city. The average was then compared to the total demand of SE2 in 2019. This resulted in a share of each city's demand of the total demand of SE2. This percentage was then used to estimate the hourly power demand for each city.

TABLE I
DATA FOR CITIES WITH MEAN VALUES FOR 2019.

Cities		
City	**PD**$_{mean}$	**Part of PD**$_{total}$
Sundsvall	600 MW	63.6 %
Sollefteå	26 MW	2.8 %
Strömsund	17 MW	1.8 %
Östersund	69 MW	7.4 %
Vilhelmina	11 MW	1.2 %
Örnsköldsvik	220 MW	23.3 %

In Table I the chosen cities with their average power demand is presented, along with a column displaying how large each city's power demand is relative to the average of the total demand of the created system.

B. Hydro Power Plants

To decide which hydro power plants from the river of Ångermanälven to include in the simulated system, all hydro power plants in the region needed to be considered.

In Fig. 10 all hydro power plants in Ångermanälven are displayed. The connecting rivers of Faxälven and Fjällsjöälven are also included.

To choose the appropriate hydro power plants, suitable for the system, their locations and maximum capacity of production needed to be considered, to match the hourly power demand. In order to avoid a large power generation deficit in the system, some larger hydro power plants were chosen over smaller ones. Additionally, only existing hydro power plants were considered since no planned construction of a new hydro power plant currently exists in Sweden [3].

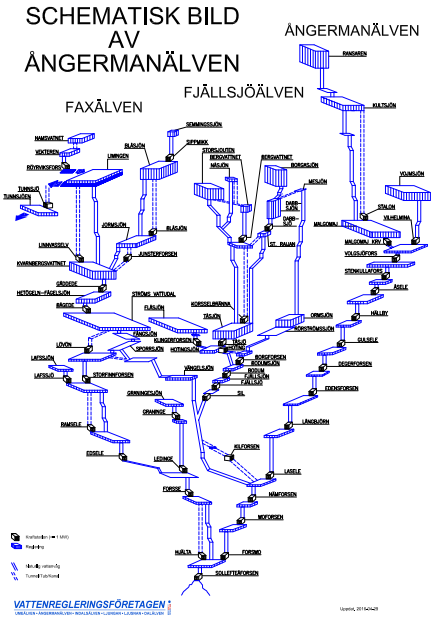


Fig. 10. Schematic figure of hydro power plants in the river of Ångermanälven, and its connecting rivers Faxälven and Fjällsjöälven [22]

Stalon, Långbjörn, Storfinnsforsen, Ramsele, Korsselbränna och Hjälta were chosen as the hydro power plants in this project. The plants were chosen as they are spread out across the river, hence not producing all the power at the same place.

The historical production data was retrieved from Svenska Kraftnät, SVK [19], and the only available data was for the hourly total production in each electricity trading area, *SE1 Luleå, SE2 Sundsvall, SE3 Stockholm, and SE4 Malmö* [20]. For each hydro power plant, the hourly power production was assigned with a percentage that represents the fraction of the total production in SE2. This fraction was the capacity for each power plant, divided by the total capacity in SE2.

The hydro power plants and its bus numbers are displayed in Table IV, while the maximum capacity of each

hydro power plant are presented in Table II. Each hydro power plant's capacity compared to the total power generation capacity of the entire designed power, and the average power generation for each hydro power plant, is also presented in Table II.

TABLE II
DATA FOR HYDRO POWER PLANTS

Hydro Power Plants			
Power plant	**Capacity**	**PG**$_{mean}$	**Part of PG**$_{tot}$
Stalon	105 MW	52 MW	6.7 %
Långbjörn	98 MW	49 MW	6.2 %
Storfinnforsen	110 MW	55 MW	7.0 %
Ramsele	155 MW	77 MW	9.8 %
Hjälta	165 MW	55 MW	10.5 %
Korsselbränna	110 MW	82 MW	7.0 %

C. Wind farms

When deciding where to put wind farms in the system constructed for simulation a lot of factors had to be considered. Firstly, the areal restrictions needed to be taken into consideration, along with areas containing unfavorable wind speed and efficiency data.

A very large part of Sweden is restricted for the construction of wind turbines and overhead transmission wires. A large amount of this restricted area is restricted due to nature conservation, such as national parks and reserves. Areas protected because of national defense interests also accounts for a large part of the restricted area [23]. However, these restrictions are not directly connected to the construction of wind turbines but shows conflict of interests for expanding wind power in Sweden. By taking this into account it was possible to exclude areas not suitable for placement of wind farms.

In addition to restrictions, the prospects of wind speed and efficiency data has also been taken into consideration. Wind data was obtained from the Swedish Energy Agency [24]. By inspecting the wind data for the annual average, it was possible to identify locations that are favorable from a production perspective. For these locations hourly production possibilities was extracted from *Renewables Ninja* [25]. Renewables Ninja allows the user to run simulations of the hourly power output from wind and solar power plants located anywhere in the world, these simulations can be extracted and processed into *.csv-files* [25], and then imported into MATLAB to be used for simulations with MATPOWER.

With this in mind, Blodrotsberget and Viksjö, two already existing wind farm parks, were selected to be included in the designed power system. Blodrotsberget is a wind farm under construction, by a collaboration between Siemens and Vattenfall Eldistribution AB, with a final capacity of 164 MW [26]. Nysäter Wind Farm is another wind farm under construction and projected to have the largest capacity for power production in Europe, and is located close to Viksjö, north of the city of Sundsvall. Its final capacity will be 475 MW [27].

With the restrictions presented in earlier in this section, Kallsjö was chosen as the third wind farm. On the one hand because the location included favorable wind speed for energy production. One the other hand because both former selected farms are on the east side of the created system, and therefore the third wind farm was placed in the west to supply the more western cities with electricity.

Kallsjö is constructed to consist of 70 wind power turbines each with a capacity of 3 MW. This results in a wind farm with a total capacity of 210 MW.

The wind farms and their location, capacity, and the average power generation efficiency are displayed in Table III. The bus numbers for the wind farms are displayed in Table IV.

TABLE III
DATA AND INFORMATION REGARDING WIND FARMS

Wind Farms			
Names	Kallsjön	Viksjö	Blodrotsberget
Coordinates: *(Lat/Long)*	63°35'10.7"N 13°9'25.2"E	62°50'6.4"N 17°18'53.7"E	63°48'37.1"N 17°57'21.6"E
Capacity: *(of PG$_{tot}$)*	210 MW (10.4 %)	475 MW (29.26 %)	164 MW (13.3 %)
$\boldsymbol{\eta}_{mean}$:	0.161	0.275	0.176

D. Transmission Grid

To connect all buses, a transmission grid was needed to be created. The base case system was created so all loads and generators were connected to each other. By attempting from the start, to ensure a base case system resilient to branch failures, it was decided that all buses should at least have two branches connected to other buses.

Since this project is not supposed to fully simulate a real existing transmission grid, the existing transmission grid in Sweden was not considered. Straight transmission lines was created between all buses, as shown in Fig. 9.

Data for these transmission lines was retrieved from Energimarknadsinspektionen [28] and for properties like resistance, reactance and susceptance, standard values were used. All branches were treated as π-equivalent line model.

Cities and wind farms were chosen as *PQ-buses*, since the active and reactive power are known for each of these buses, as the used data is based on historical data for active power demand and production. The reactive power for the buses was then calculated by using a specific power factor set to 0.98. The power factor indicates how much power is being used to perform useful work, in other words it is the ratio between the active power P and the apparent power S. More on this in section *II-B. Load flow study.*

Hydro power plants were chosen as *PU-buses*, due to the fact that the active power generated and voltage is known for all six of the hydro power plants.

The *slack bus* was placed in the northeast to represent a connection to the northern transmission grid in SE1, since there is a constant power deficit in the system.

Typically, SE1 and SE2 have higher power generation than demand, and therefore export to more southern parts of the transmission grid [20]. Since most of the power generation is in the northern part of Sweden [29], it was decided that the created system will import power from SE1 in the north.

TABLE IV
System explanation with bus numbers and correlating bus names and types

Bus number:	***City:***	***Bus type:***
1	Sundsvall	PQ
2	Sollefteå	PQ
3	Strömsund	PQ
4	Östersund	PQ
5	Vilhelmina	PQ
6	Örnsköldsvik	PQ
	Hydro Power Plant	
7	Stalon	PU
8	Långbjörn	PU
9	Storfinnsforsen	PU
10	Ramsele	PU
11	Korsselbränna	PU
12	Hjälta	PU
	Wind Farm	
13	Blodrotsberget	PQ
14	Viksjö	PQ
15	Kallsjön	PQ
	Slack bus	
16	Eastern node	Slack

E. Simulation period

To ease display and analysis of data and graphs, four weeks were chosen for simulation. Two of these weeks are based on wind power generation; one week for maximum production and one week for minimal production. The remaining two weeks were chosen to include one week with maximal demand in the loads, *cities*, and one week containing a mix of low power generation from the wind farms, with high demand. The chosen weeks are displayed in Table V. By accident, the week for minimal wind production aligned with the week with minimal power demand.

TABLE V
Weeks chosen for analysis

Weeks analyzed (2019)	
Maximum wind production	Week: 12 (March 18 - March 24)
Minimal wind production + minimal power demand	Week: 30 (July 22, 2019 - July 28)
Maximum power demand	Week: 5 (January 28 to February 03)
Low wind production + high power demand	Week: 41 (October 07 to October 13)

F. Case analysis

To determine whether the created system was working within the set limit of a maximum of 10 % deviation from the set base voltage, a snapshot of the system was taken every hour of the analyzed time-period. System conditions were created for a specific point in time, a snapshot, and the values within the system at this snapshot were then used for analysis. With one snapshot every hour, for every week, the result was 168 snapshots per analyzed week. Values for all power flow, generation, and demand; currents; voltage magnitudes and angles; and much more; were calculated for the entire system, at every snapshot.

To assess the quality of the created and simulated system, some cases were decided to be investigated. The main goal was to design a system that would stay within 10 % voltage deviations at every node at every time.

For this a base case, with a base voltage set to 220 kV, was created. When analyzing the results of the simulations done with the base case, limits regarding the voltage deviations was looked at.

Svenska kraftnät states that *to replace one 400 kV transmission line, would require four to eight 220 kV lines* [30]. To assess how an increased base voltage would improve the designed grid, a further investigation of changing the base voltage was therefore conducted, even if the voltage deviation goal was already fulfilled.

To make this analysis possible, two more cases were created. One case consisting of a grid with a base voltage set to 245 kV, and as seeing that the newer voltage magnitude for long distance power transmission is 400 kV in Sweden, a case was created with a base voltage set to 400 kV.

In the additional cases, voltage deviation and losses were compared for the different cases. This was conducted to investigate how obtain lower losses in the system and, at the same time, increase the transmission capacity.

Even though faults on the overhead transmission are usually rectified within 24 hours, unreliable power production can lead to costly power outages [5]. To avoid costs connected to faults in the transmission grid, a more resilient transmission grid might be needed, and as stated by the set goals, a certain resiliency in the transmission grid was sought. Therefore, two additional cases were created. One for the N-1 criterion and one for the N-2 criterion.

1) Base case: The base case consists of the grid presented in Fig. 9. That grid contains 28 branches of various lengths, 10 generator buses and 6 loads. The base voltage is set to 220 kV. The transmission lines used for the analysis are based on lines with a capacity 245 kV. Although this is the case, the base voltage was set to 220 kV, since this is the older nominal voltage in Sweden for the long-distance transmission [31].

The simulated system utilizes 220 kV as base voltage for the base case. The grid therefore needed to be dimensioned for a minimum of 220 kV to be able to function correctly. The costs for a complete three-phase *(FeAl)* overhead wire dimensioned for 245 kV per kilometer is 2 583 053 SEK/km for a cross section of 910 mm^2 [28]. No data is available for transmission lines with a capacity of exactly 220 kV, and therefore a transmission grid was chosen with a capacity

similar to the desired one, with the closest match being 245 kV.

2) 245 kV-case: Considering that the transmissions lines used for analysis in the base case have a capacity of 245 kV, a 245 kV-case was also created. In this case the base voltage was set to 245 kV, with everything else the same.

3) 400 kV-case: The base voltage was set to the new higher 400 kV. With the new higher voltage exceeding the capacity of 245 kV, new line data was needed to be calculated.

As for a 400 kV overhead wire, the mean cost is estimated to be 598 231 euro/km [32]. With the exchange rate 10.19 SEK/euro the cost in SEK is 6 095 974 SEK/km.

4) N-1 case: To simulate the N-1 criterion, one branch at the time was disconnected and system simulations was done without the disconnected branch. The time period for the N-1 simulations were only one day, compared to the other simulations that were done over a four-week period. This day was chosen as the day with the largest deviation, in the base case, of all analyzed days. If the system could keep the voltage deviation under 10 %, the system was considered reliable. However, if the voltage limitation was violated when single a branch was disconnected, an in-depth analysis was made how to remedy this violation.

5) N-2 case: For the N-2 criterion, only branches connected to Sundsvall and the slack bus was examined. Sundsvall was selected as a critical area as it is the largest city in the analyzed system, as well as the largest consumer in the system. The slack bus imports power from the north to cover any power deficit in the system and is therefore also crucial. The N-2 criterion was tested in the same way as the N-1 criterion. If the system was not able to withstand two components failing, additional branches were installed between various buses until the system is fulfilling the requirements for resiliency.

V. Results

The results simulated in MATLAB with MATPOWER will be presented in this section. The results will include the base case system's voltage deviation for the four weeks analyzed. Additionally, graphics that show demand, production and voltage deviation will be included.

Furthermore, development of the base case system will be presented. The developments will consist of the results from the 245-kV case and 400 kV-case as well as from the N-1 and N-2 criterion simulations. Also, the economic cost of upgrading the base system will be analyzed.

An additional result that will be analyzed, is the losses in the different cases. The base case system's losses will be compared to the upgraded systems.

A. Voltage magnitude deviations

After analyzing the selected four weeks, the voltage deviation was kept within 10 % for all buses in the base-case system. The maximal voltage deviation was 6.93 % in Sundsvall (bus 1). In Fig. 11, the voltage magnitude for

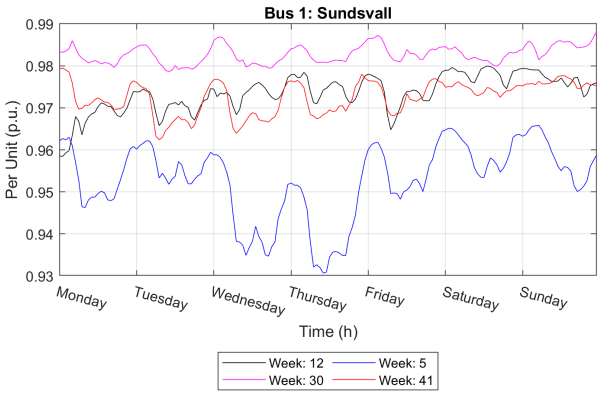


Fig. 11. Voltage magnitude (p.u.) for bus 1, the city of Sundsvall, which displays the bus that has the largest deviation every selected week. Every day consists of 24 data points.

Sundsvall is shown. From this figure, it is clear, that the week with largest deviation from the reference value was week 5. Maximum power demand, and the maximal voltage deviation occurred on Thursday, which was January 31st.

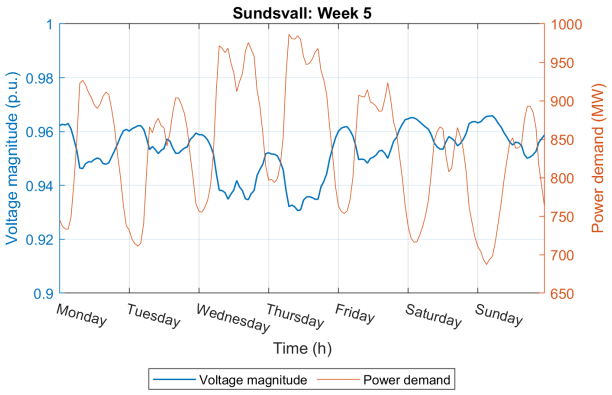


Fig. 12. Voltage magnitude (p.u.) and power demand (MW) in Sundsvall.

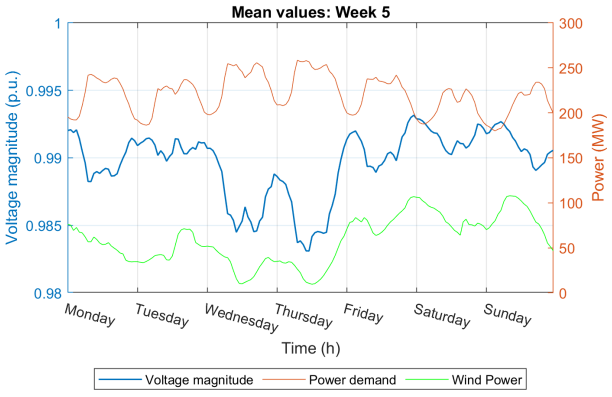


Fig. 13. Mean values for all voltage magnitudes (p.u.), power demands (MW) and power generated (MW) from wind in the entire system. The power demand (orange) and the power generated from wind (green) share the right y-axis.

The reason for the maximal voltage deviation occurring

when it did was also examined. A comparison was made between the voltage magnitude and power demand in Sundsvall for week 5. The result is shown in Fig. 12. This figure shows that there is an increase in power demand when the voltage magnitude decreases. When the power demand is lower, the voltage magnitude increases. However, when the demand is at it lowest on Sunday, the voltage magnitude is approximately the same as on Saturday, but Saturday has a higher power demand.

In Fig. 13 mean values for all buses' voltage magnitudes are shown, as well as the mean value of the power demand and generated power from wind power in the entire system. The results showed that Thursday, January 31st, during week 5, was the day with not only the maximal mean voltage deviation bus also had the highest value for power demand and the lowest value for power generated from wind power. This shows that the voltage magnitude does not only correlate with power demand, but also with the current power generation.

B. Results of simulations

1) Base case: The voltage magnitude deviations were kept within the allowed limits, as shown in Fig. 11. This result is also shown in Table VI.

The total length of the three-phase transmission lines in the entire simulated transmission grid summed up to 2067 kilometers. To implement this grid, dimensioned for 245 kV, the cost would be about 5.3 billion SEK. The grid was dimensioned for 245 kV, and not 220 kV, due to the lack of data for transmission lines with a capacity of 220 kV.

TABLE VI
Base case results

Base case (220 kV)				
Week:	***12***	***30***	***5***	***41***
Max deviation (%)	4.15	2.13	6.93	3.76
Mean deviation (%)	0.49	0.34	1.04	0.57
Total losses / demand (%)	1.44	1.21	2.56	1.81

2) 245 kV-case: The result for when the base voltage was increased to 245 kV is shown in Table VII. By increasing the base voltage, the maximum mean voltage deviation decreased from 6.93 % to 4.69 %. For all four weeks analyzed the mean deviation, as well as the mean losses, decreased in the system when increasing the base voltage.

As the overhead wires selected for the base case are designed for a maximum voltage of 245 kV, it would be possible to increase the voltage to its capacity at 245 kV. This results in less power losses in the grid, as well as smaller voltage deviations.

3) 400 kV-case: The result for when the base voltage was increased to 400 kV is shown in Table VIII. The maximum voltage deviation for all four weeks is around, or below 1 %. For the mean deviation, the voltage was kept very close to the base value, with a mean deviation of lower than 0.15 %. The losses decreased for the 400 kV case for all weeks compared to the losses in the base case.

TABLE VII
245 kV-case results

245 kV-case				
Week:	***12***	***30***	***5***	***41***
Max deviation (%)	2.75	1.60	4.69	2.65
Mean deviation (%)	0.33	0.25	0.68	0.40
Total losses / demand (%)	1.14	0.97	2.00	1.44

The cost of implementation and construction of a transmission grid dimensioned for a capacity of 400 kV, would be around 12.6 billion SEK.

TABLE VIII
400 kV-case results

400 kV-case				
Week:	***12***	***30***	***5***	***41***
Max deviation (%)	0.72	0.49	1.08	0.73
Mean deviation (%)	0.08	0.07	0.15	0.11
Total losses / demand (%)	0.42	0.36	0.71	0.52

TABLE IX
Comparison between different cases and their maximum values for all weeks.

Comparison			
Case:	***220 kV***	***245 kV***	***400 kV***
Max deviation (%)	6.93	4.69	1.08
Mean deviation (%)	1.04	0.68	0.15
Total losses / demand (%)	2.56	2.00	0.71

A comparison between the three cases are made in Table IX. The results show that increased base voltage results in decreased of deviations and losses. This result was to be expected, as in section *II-E. Power losses in the transmission system*, it was explained that losses should decrease with increased voltage.

Fig. 14 is a more graphical way of presenting the results from Table IX. It is clear, that a 400 kV transmission grid dramatically reduces both the losses in transmission, as well as the voltage deviations.

C. Resiliency results

1) N-1 criterion: The base case system could withstand branch failure for one branch at the time for all branches except one. The branch that could not be disabled without breaching the safety limitations in the system was the branch between the Slack bus and Örnsköldsvik. As show in Fig. 16 with the orange arrow. A failure on the branch caused instability in the entire system and the voltage deviation limitations were violated.

To minimize the impact of a failure between the Slack bus and Örnsköldsvik, an additional branch was added to the system. A branch between Slack bus and Sundsvall was deemed suitable, as after implementation the voltage limitations were no longer violated when the branch between Slack bus and Örnsköldvik was disconnected. Hence the N-1 criterion was fulfilled, and resilience achieved. The

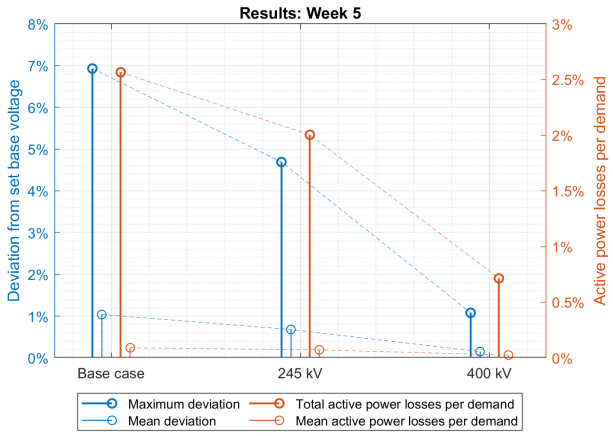


Fig. 14. Stem plot showing how an increased base voltage decreases both deviations and losses. The orange stems are losses per demand. The thicker stem is the total active power losses during week 5 as a percentage of the total demand that week. The thinner stem is the mean active power losses during week 5 as a percentage of the hourly mean demand of the entire system that week. (Please note that there are two different y-axis)

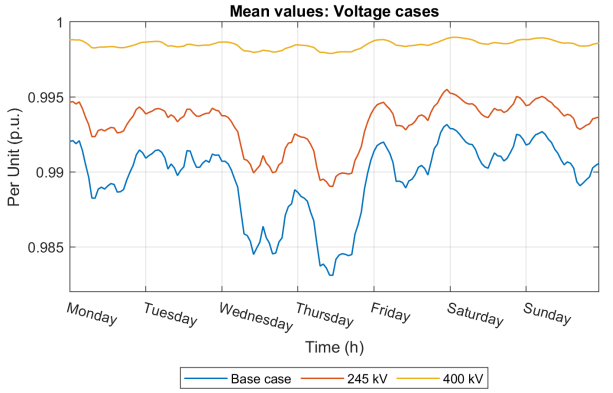


Fig. 15. Mean per unit values for the voltage magnitudes of all buses in the entire system during week 5 for all voltage cases.

system which fulfills the N-1 criterion is shown in Fig. 16 with the added transmission line in blue.

2) N-2 criterion: Only Sundsvall and the Slack bus were considered for the N-2 criterion. Since the created system was not able to fulfill the N-1 criterion without installing an additional transmission line between Sundsvall and the Slack bus, this transmission line was deemed necessary to be included when analyzing the resiliency according to the N-2 criterion. This transmission line can be seen by the blue N-1 arrow in Fig. 17.

For the N-2 criterion, several combinations of branches were problematic when two failed at the same time, even though the breeches of the limitation were not as serious as the ones with the N-1 criterion. Therefore, as an addition to the branch already added during the N-1 simulations, an additional branch was connected, one at a time, between Sundsvall and multiple different buses.

One transmission line that was deemed not to be as important as the other ones, was the transmission line between Sundsvall and Östersund, referred to with the

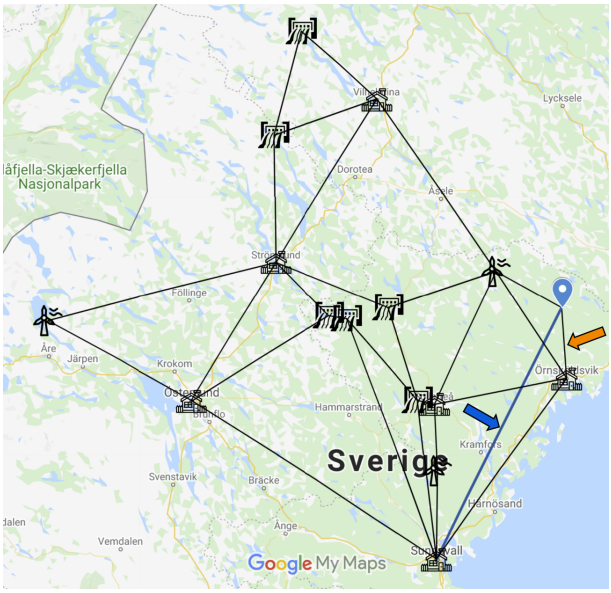


Fig. 16. Overview of system with regard to the N-1 criterion.

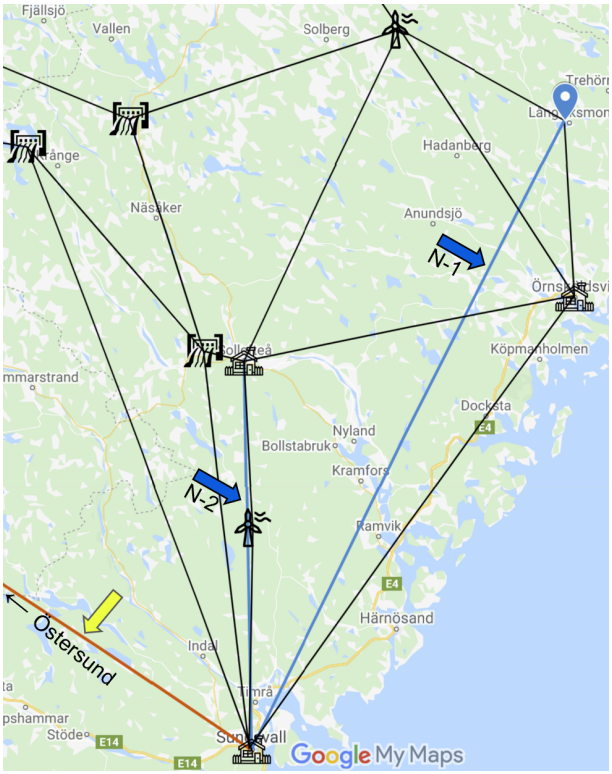


Fig. 17. Overview of system with regard to the N-2 criterion.

yellow arrow in Fig 17. It was possible to keep voltage deviations within the limit of 10 % when this branch was disconnected at the same time as another branch from Sundsvall, or the Slack bus.

The most suitable branch was found to be between Sollefteå and Sundsvall, as shown in Fig. 17 with the second blue arrow. The branch made sure that the voltage deviations would not exceed the limit of 10 % when two branches, in a crucial area, were disconnected. The

selected branch had the shortest possible length of transmission lines needed to fulfill the N-2 criterion and was therefore a more economical choice than the other options.

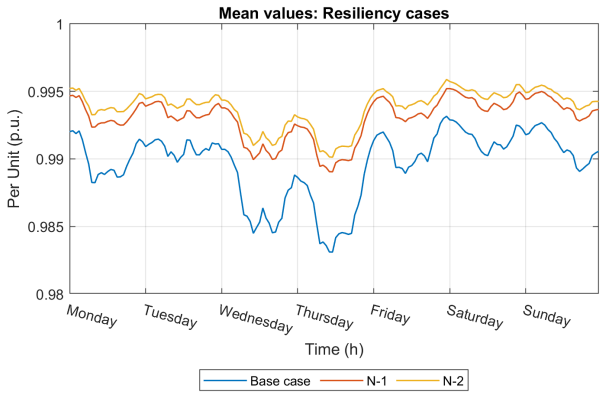


Fig. 18. Mean per unit values for the voltage magnitudes for the entire system during week 5 for the base case, as well as the different resiliency cases.

Installing additional transmission lines in order to improve the resiliency of the grid also improved the grid in other ways, such as decreased voltage deviations. This is presented in Fig. 18, where the mean values for the voltage magnitudes are displayed for the base case, as well as the two resiliency cases.

VI. Discussion

Considering how an increased amount of wind power would affect the voltage in a real power system, a lot of different aspects must be considered. Voltage deviations need to be analyzed, at the same time as socio-technical systems and profitability assessments are made.

All the while the result of this study was desirable and the goals were met, the simulation and analysis must be made more extensive to get a better understanding of what would happen in specific areas.

A. Power system

A very basic, relatively speaking, transmission grid has been simulated in this study. While this is the case, the general simulations have resulted in a concept and an idea that points in a good direction.

A lot of inputs were overlooked in the modeled transmission grid, and to extend this study to enable it to become a basis for future decisions, these overlooked inputs must be looked at.

The first identified input that was overlooked, is the fact that the simulations made in this project does not take the real, already existing, transmission grid into account. There are a lot of additional buses, such as cities and power stations, that were not included. By looking at Fig. 10, it is clear that only a fraction of the hydro power plants included in the schematic figure was used in the simulated grid. The same goes for the wind farms, there are a lot of existing wind farms not included in the study. The result is therefore not applicable to the real area around Ångermanälven, and the results will, most definitely, differ from the real voltage levels.

The entire grid has been considered to only consist of the backbone of the grid, with one unified voltage level. Considering that the real transmission grid consists of a lot of different voltage levels, this might have to be included to deem whether it is important for the results.

To present reliable results about a 400 kV grid and if it is worth investing in, the simulations should include accurate data of the characteristics of the transmission lines. There is no data available for the 400 kV lines, and therefore the data used are standard values that might be different from actual 400 kV transmission lines.

Therefore, to create a more realistic scenario, the existing transmission grid in the area could have been implemented, as well as more buses. However, the real existing transmission grid was outside the scope of this project. This is because the goal of the project was not to simulate the real grid, but to try to analyze how the transition to more wind power might affect voltage levels. For future purposes it would be interesting to implement the methods used in this project to investigate how additional wind power might affect existing power systems.

When implementing the results in a real power system, there are a multiple factors in play. Not only the cost of the actual construction, but also the fact that regulations exist that prohibits transmission lines in certain areas. The straight transmission lines drawn between every node in this study might therefore not even be possible due to restrictions. The regulation of high voltage power lines is also outside the scope of this project.

B. Aspects of wind farms

When deciding for where to simulate a large wind farm, a conclusion was made that the wind farms should not be placed too far away from cities, roads and existing transmission lines as this would result in greater costs for constructing infrastructure.

As presented in the results, it is possible to generate power from wind farms without breaching the safety limitations of the system. However, the placement of wind power could be problematic. As shown in Fig. 13, it is possible for the system to keep the voltage within allowed limits even when the wind generated power production decreased. Therefore, it was possible to keep the stability in the system even though a remote wind farm was placed in Kallsjö, far from the biggest consumers. For further purposes, an investigation could be made to examine the significance of wind farm placement and how it may affect voltage deviations.

For instance, an in-depth study could be made regarding wind farms placements and how they might compromise the stability of the grid. This could be done in a general case, and hopefully identify characteristics of what scenarios that should be avoided. This could aid tech-oriented decision-makers when planning wind farm construction. There is a great deal of restrictions that makes it difficult

to obtain permits for wind farms, this along with the study regarding placements, could be an incentive to study different alternative placement of the wind farms in further studies.

In section *VI-A. Power system*, the fact that a lot of existing wind farms was not included in the designed system, was explained. Another input that was overlooked was that different models and the hub height will affect the efficiency and power generation of the wind turbines. A future study might therefore investigate where different turbines are most optimally placed, when both considering political and technical views. These political views might go into the often talked about subject called NIMBY, *not in my backyard*. The future study might result in a suitable way to weigh the favorable and unfavorable factors about where different wind power turbines should be optimally placed and constructed.

C. Resiliency in the grid

The N-1 criterion is a must for a real power system, and the implementation of an additional transmission lines were a necessity to accomplish resiliency in the simulated system. However, it was not investigated if the transmission lines of the base case system, shown in Fig. 9, was properly designed. It may have been possible to design a more resilient system, consisting of fewer meters of transmission lines, and still avoiding the instability issue arising when having a failure between the Slack bus and Östersund. A grid consisting of the least amount of transmission lines, but still being resilient, might be more cost-effective. The same argument goes for the N-2 criterion, rearranging the connections from Sundsvall could lead to a more cost-effective way of designing the system.

The additional branch between Slack bus and Sundsvall did not only fix the instability problem to fulfill the N-1 criterion, but the voltage deviations was also improved, shown in Fig. 18. Furthermore, the voltage deviation decreased when implementing the additional branch from the N-2 criterion. Keeping the limits within a maximal voltage deviation of 5 %. However, additional branches are costly to implement and should therefore be avoided to some extent. For more critical parts of systems that do not allow voltage deviations up to 10 %, it is great to keep in mind that one branch could be enough to remedy exceeding voltage deviations within the system.

D. Profitability assessment

There is need to utilize the transmission grid in the most cost-efficient way and still meet the power demand of all consumers. Therefore, it is of utter importance to choose a suitable nominal voltage for the transmission grid that can fulfill these criteria.

With large production capacity in the northern parts of Sweden, and high demand for electricity in the south. A bottleneck occurs when the transmission capacity is lower than the desired demand and is often an issue during peaks in demand. To avoid these bottlenecks in the grid, the transmission grid needs to be expanded. As presented in section *V-B. Results of simulations*, a transmission grid with a capacity of 220 kV is significantly cheaper than a grid consisting of 400 kV transmission lines.

However, a 400 kV grid is favorable as the capacity for transmission is higher than 220 kV, along with the fact that losses are dramatically decreased when a higher voltage level is used. This is also presented in section *V-B. Results of simulations*. Therefore, to meet the ever-increasing power demand in Sweden, a 400 kV transmission grid would be more suitable than a 220 kV transmission grid, despite being more expensive.

Looking at the results in Table IX, it is clear that the maximum and mean voltage deviations are more than six times larger in the 220 kV-case compared to the 400 kV-case. Using this result with the example by Svenska kraftnät, presented in section *IV-F. Case analysis*, would mean that the cost of six 220 kV systems (like the one in Fig. 9) would be about $5.3 \times 6 = 31$ billion SEK compared to 12.6 billion SEK for the 400 kV grid.

A 400 kV grid could also contribute to a great deal of socioeconomic benefits as industries could expand their businesses, and the grid reliability would be increased to meet said demand. A good example of this is looking at Fig. 15. An upgrade to 400 kV, indicates that the power system would have very small voltage deviations and lower losses. Hence, a 400 kV transmission grid hopefully contributes to economic growth and the benefits in social welfare might be higher than the cost of the investments, since the grid is more resilient and will be able to meet an increase in demand.

The voltage magnitude decreases when the demand increases as shown in Fig. 12. Considered that a base voltage of 400 kV results in voltage deviations below 1 % the demand could increase significantly and still be within the limits compared to the base case's maximum deviation of 6.93 %.

Furthermore, the losses decreased from 2.56 % for the base case to 0.71 % for the 400 kV case, as shown in Table VIII. Additionally, the cost for the 400 kV case was 7.3 billion higher than the base case. This results in a cost of 3.9 billion per percentage of decreased losses. However, as mentioned before, the decrease in voltage magnitude has to be taken into account when analyzing if a higher base voltage is profitable. Investing proactively in a transmission grid will allow expansion and is perhaps more favorable even though the cost is significantly higher than designing a grid that is sufficient for the existing load. Lower losses result in more power available to meet an increase in demand and still lowering the voltage deviations. As argued earlier, this could result in the possibility to meet higher demand than before and still keep the voltage deviation within the limitations.

E. Power flow

Since most of Sweden's electricity is being generated in the north, there is typically power being transferred

from the north to the south. Therefore, the electricity trading area SE2, is usually exporting power to the more southern trading areas, SE3 and SE4. In this study, this is not an option since there is no simulated connection to the south. However, even with an included slack bus with a maximum power generation of zero (power generated below zero equals export), this would not have happened in the simulations. The reasons being, that the included consumers are much larger than the included producers. This results in that the created system is in constant demand of power and must import power from the Slack bus. This is also the reason for the fact that the voltage deviations are never larger than the set base value. The voltage levels are always below 1 per unit.

By including a larger share of power plants in the region, as well as adding a second slack bus for the export in the south, the simulated system would most likely have been able to export power. Another probable result of including more power plants would be voltage magnitude exceeding 1 p.u. at some point.

F. MATPOWER and MATLAB calculations

As presented in section *III. MATPOWER*, a setting that enforced limits in the reactive power generation was activated. Without enforcing limits in the reactive power, the generator buses that was set to PU-buses (i.e., hydro power plants) was generating an unrealistic amount of reactive power. Seeing as the slack bus was considered a connection to another transmission grid, it was deemed better to get the needed reactive power to solve the power flow from the Slack bus, than having unrealistic generators. Although the Slack bus was injecting a lot of reactive power into the grid, the voltage deviations took a hit after enforcing the power limits, resulting in twice as much deviations as before the limitations. This setting also resulted in that the five out of six generator buses that was set to PU-buses, were converted to PQ-buses.

VII. Conclusion

In this project a power system was designed and modeled in MATLAB. The designed transmission grid consists of hydro power plants, cities, wind farms and transmission lines. With real historical data for production, demand, and wind speeds, hourly snapshots of the load flow was simulated using MATPOWER.

It was then possible to use this system to simulate how, and why, voltage deviations are affected. Implementing wind farms in this modeled transmission grid, creates opportunities to analyze how these might affect voltages in a real transmission grid, and by simulating connections to other areas with a slack bus, it is possible to meet power demand, even when there exist a power deficit in the grid, just like in a real transmission grid.

Even with this modeled power system, with a base voltage set to 220 kV and wind power standing for 53 % of the total power generation capacity, was it possible to keep the voltage deviations below the limit of 10 %.

When considering voltage deviations, losses, and resiliency, it was possible to create a far superior transmission grid by increasing the voltage and redesigning the grid.

When increasing the base voltage to 400 kV, the voltage deviations was kept within 1 %, and by installing two additional branches to the grid both the N-1 and N-2 resiliency criteria were fulfilled, as well as lowering the voltage deviations further.

The drawback of increasing the base voltage, is the increased costs for constructing the system. These additional required investments were compared to advantages of the 400 kV power system. One of these advantages were the significantly decreased losses in the system. The conclusion was that the increased cost for a 400 kV system is a profitable investment in the long run, as it creates a more reliable grid, with an increased transmission capacity, which will be needed in the ever more electrified society.

Appendix A
MATPOWER structure array

Acknowledgment

We would like to thank our project supervisors Evelin Blom and Lennart Söder for the help and wonderful insight during this project. We would like to express an additional gratitude to Evelin and her willingness to aid our progress, from the very start to the very end. Without your extraordinary helpful attitude and deep knowledge in the field, this would not have been possible.

References

[1] Statistikmyndigheten, SCB. (2021, Feb) Elektricitet i Sverige - Vindkraftsproduktion 2000-2019 (GWh). [Online]. Available: https://www.scb.se/hitta-statistik/sverige-i-siffror/miljo/elektricitet-i-sverige/

[2] N. Boccard, "The cost of nuclear electricity: France after fukushima," *Energy Policy*, vol. 66, pp. 450–461, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0301421513011440

[3] K. Byman, "Sveriges framtida elproduktion - en delrapport - iva-projektet vägval el," Kungl. Ingenjörsvetenskapsakademien (IVA), Stockholm, Sweden, Tech. Rep., 2016. [Online]. Available: https://www.iva.se/globalassets/info-trycksaker/vagval-el/vagvalel-sveriges-framtida-elproduktion.pdf

[4] Svenska Kraftnät, SVK. (2015, Dec) Anpassning av elsystemet med en stor mängd förnybar elproduktion - en slutrapport från Svenska kraftnät. Sundbyberg, Sweden. [Online]. Available: https://www.svk.se/siteassets/om-oss/rapporter/anpassning-av-elsystemet-med-en-stor-mangd-fornybar-elproduktion.pdf

[5] ——. (2014, May) Elnät i fysisk planering - behandling av ledning och stationer i fysisk planering och i tillståndsärenden. Sundbyberg, Sweden. [Online]. Available: https://www.svk.se/siteassets/om-oss/rapporter/elnat-i-fysisk-planering-webb.pdf

[6] MATPOWER. (2021, Apr) Free, open-source tools for electric power system simulation and optimization. [Online]. Available: https://matpower.org

[7] MathWorks. (2021, Apr) MATLAB is a programming and numeric computing platform used by millions of engineers and scientists to analyze data, develop algorithms, and create models. [Online]. Available: https://www.mathworks.com/products/matlab.html

[8] L. Söder and M. Ghandhari. (2015, Aug) Static Analysis of Power Systems. [Online]. Available: https://www.kth.se/social/files/55f17d7df2765458ad9c151b/comp_eg2100_ht15_v2.pdf

[9] M. Mostafa, M. Anwar, and A. Radwan, "Application of electrical resistivity measurements as quality control test for calcareous soil," *Housing and Building National Research Center*, vol. 14, no. 3, pp. 379–384, July, 2017.

[10] L. Söder. (2005, Jan) Statisk Analys av Elsystem. [Online]. Available: https://kth.diva-portal.org/smash/get/diva2:1075656/FULLTEXT01.pdf

[11] H.-P. Nee, M. Leksell, S. Östlund, and L. Söder, *Eleffektsystem*. Stockholm, Sweden: KTH, Royal Institute of Technology, 2019, pp. 2–5–2–6.

[12] N. Kang and Y. Liao, "Equivalent pi circuit for zero-sequence double circuit transmission lines," in *2012 IEEE Power and Energy Society General Meeting*, 2012, pp. 1–6.

[13] L. Råde, B. Westergren, and F. Wikström, *Beta - Mathematics handbook - for scienece and engineering*. Lund, Sweden: Studentlitteratur AB, 2019, p. 418.

[14] "Ieee recommended practice for conducting load-flow studies and analysis of industrial and commercial power systems," *IEEE Std 3002.2-2018*, pp. 1–73, 2018.

[15] Svenska kraftnät, SVK. (2009) Stamnätets tekniskt-ekonomiska dimensionering. [Online]. Available: https://www.svk.se/siteassets/om-oss/rapporter/091202_stamnatets_dimensionering_aterrapport.pdf

[16] R. D. Zimmerman and C. E. Murillo-Sánchez, *MATPOWER User's Manual*, 7.1 ed., Power System Engineering Research Center (PSerc), Tempe, AZ, Oct 2020.

[17] Google. (2021, Mar) MyMaps - D1: Voltage by Klas and Sophia. [Online]. Available: https://www.google.com/maps/d/u/0/edit?mid=1IE8Ym2jC4-VuK1smJFyjJc4xYK4CAc8I&usp=sharing

[18] SCB. (2021, Apr) Elproduktion och bränsleanvändning (mwh), efter län och kommun, produktionssätt samt bränsletyp. År 2009 - 2019. [Online]. Available: http://www.statistikdatabasen.scb.se/pxweb/sv/ssd/START__EN__EN0203/ProdbrEl/

[19] Svenska kraftnät, SVK. (2021, Apr) Elstatistik. Sundbyberg, Sweden. [Online]. Available: https://www.svk.se/om-kraftsystemet/kraftsystemdata/elstatistik/

[20] Energimarknadsinspektionen. (2021, Apr) Elområde. [Online]. Available: https://www.ei.se/konsument/el/sa-har-fungerar-elmarknaden/elomrade

[21] Nord Pool. (2020) Electricity spot price areas - bidding areas. [Online]. Available: https://www.nordpoolgroup.com/the-power-market/Bidding-areas/

[22] Vattenregleringsföretagen. (2016, Apr) Schematisk bild av Ångermanälven. [Online]. Available: https://www.vattenreglering.se/wp-content/uploads/AVF_schematisk_bild.pdf

[23] Statens energimyndighet, *100 procent förnybar el*. Bromma, Sweden: Arkitektkopia AB, 2019, pp. 20–23.

[24] H. Bergström and S. Söderberg. (2012, Mar) Beräkning av vindklimatet i sverige med 0,25 km^2 upplösning med hjälp av MIUU-modellen. [Online]. Available: https://www.energimyndigheten.se/globalassets/fornybart/framjande-av-vindkraft/vindkartering/slutrapport-uppdatering-av-vindkarteringen.pdf

[25] Renewable.ninja. (2021, Mar) Simulations of the hourly power output from wind and solar power plants. [Online]. Available: https://www.renewables.ninja/

[26] Siemens Gamesa. (2019, Nov) Clean, green search for Google´s users, as Siemens Gamesa's wind turbines in Sweden will power its data center. [Online]. Available: https://www.siemensgamesa.com/en-int/newsroom/2019/11/191107-siemens-gamesa-stavron-sweden

[27] RWE. (2021) Onshore windfarm - Nysäter - fakta och siffror. [Online]. Available: https://se.rwe.com/lokaliseringar/nysaeter-onshore-windfarm

[28] Energimarknadsinspektionen. (2015, Feb) Normvärdeslista elnät 2016-2019. [Online]. Available: https://www.energimarknadsinspektionen.se/Documents/Forhandsreglering_el/2016_2019/Dokument/1/Normvardeslista_elnat_2016-2019.pdf

[29] ——. (2019, Mar) Här sker elproduktion och elanvändning i Sverige. [Online]. Available: https://www.energimyndigheten.se/globalassets/om-oss/lagesrapporter/elmarknaden/2019/mars/har-sker-elproduktion-och-elanvandning-i-sverige.pdf

[30] Svenska kraftnät, SVK. (2021, Feb) Teknik - ett svenskt transmissionsnät med växelström. Sundbyberg, Sweden. [Online]. Available: https://www.svk.se/utveckling-av-kraftsystemet/transmissionsnatet/utbyggnadsprocessen/teknik/

[31] Energimarknadsbyrån. (2020, Mar) Elens väg - stamnät. [Online]. Available: https://www.energimarknadsbyran.se/el/elmarknaden/elnatet/elens-vag/

[32] Agency for Cooperation of Energy Regulators. (2015, Aug) Electricity infrastructure. [Online]. Available: https://www.acer.europa.eu/Official_documents/Publications/UIC_Electricity_History/UIC%$20report%20%20-%20Electricity%20Infrastructure%20corrected.pdf

Implementation of a Capacity Market in Sweden

Jakob Björns and Erik Lindberg

***Abstract*—In the coming decades there will be an increase of electricity consumption as the industry and transportation sectors are electrified. Electrification and the rapid expansion of renewables will have great impact on the electricity market. In order to ensure that there is enough electricity, different capacity mechanisms are possible solutions. In this paper, the need of a capacity market in Sweden is studied. Simulations of capacity markets with convex downwards sloping demand curves demonstrated possible outcomes of such a market. A comparison between a FCM and Sweden's strategic reserve is also carried out. The results of this project show that even though peaking power plants are not profitable, there is no immediate need for a forward capacity market.**

***Sammanfattning*—De kommande årtiondena kommer elförbrukniknen att öka till följd av elektrifieringen av industri- och transportsektorn. Elektrifieringen och skiftet till förnybara energikällor kommer ha en stor påverkan på elmarknaden. För att se till att det finns tillräckligt med effekt kan en framåtblickande kapacitetsmarknad behövas. I detta projekt undersöks behovet av en svensk kapacitetsmarknad. En modell för kapacitetsmarknader med en konvex nedåtlutande efterfrågekurva gjordes för att undersöka möjliga utfall. En jämförelse mellan en framåtblickande kapacitetsmarknad och den svenska effektreserven genomförs också. Resultaten i detta arbete visar att även fast spetskraftverk inte är lönsamma så finns det inget omedelbart behov av en framåtblickande kapacitetsmarknad.**

***Index Terms*—Forward capacity market, Value of lost load, Cost of new entry, Capacity demand curves, Loss of load expectation**

Supervisor: Mohammad Reza Hesamzadeh

TRITA number: TRITA-EECS-EX-2021:155

ACRONYMS

LOLE	Loss of load expectation
CONE	Cost of new entry
VOLL	Value of lost load
SO	System operator
NE	New England
ISO-NE	The system operator in New England
ICZ	Import constrained zone
ECZ	Export constrained zone
FCM	Forward capacity market
LSE	Load serving entity
PJM	The system operator in Pennsylvania
MRI	Marginal reliability impact
EUE	Expected unserved energy

I. INTRODUCTION

Most of the world's nations has committed to limit global warming to 1.5 degrees Celsius through the Paris agreement [1]. To achieve this goal, sectors emitting large amounts of greenhouse gases like industry and transport need to be electrified. At the same time, the electricity generated must be shifted to low carbon sources. The shift includes renewables, nuclear energy and using gas instead of coal (gas emits less CO2 than coal). Cheap, renewable energy is making gas and nuclear energy unprofitable. Transitioning to a CO2 free power system will pose challenges in the design of the power system [2]. The potential loss of energy sources where the production can be controlled, could significantly decrease the reliability of the power system.

A possible solution to this problem is the implementation of an additional market for capacity. Power plants in traditional markets receive payments for energy only. In capacity markets however, the power plants receive extra payments based on the capacity they have available aside from their energy market revenue [3].

A. Background

Since the 1990s, different types of capacity markets have been implemented. The problem that capacity markets aimed to solve was the missing money problem, caused by regulatory interventions. This insufficient return on investment, results in suppliers choosing not to invest in new generation. As a result, the reliability of the power system will not meet the criteria set out by the policy makers. Capacity markets have constantly been under development since new regulations are enforced. Today however, the large share of weather dependent energy sources is responsible for the missing money problem [4].

Existing literature covering the implementation on capacity markets primarily covers the markets in U.S. In [5], a new way of deriving demand curves used in forward capacity markets for the New England power system is presented. The demand curves presented have two components in order to account for different constraints in transmission areas. Building new capacity has a different impact on reliability, depending on where there is a transmission bottleneck. Reference [6] covers how different versions of capacity markets in PJM were designed and how they have performed in the past. The paper concludes that the intended purpose of solving the missing money problem was fulfilled.

The purpose of this project is to examine the need for a capacity market in Sweden and compare different ways to set up a capacity market. In Sweden, nuclear energy provides a predictable amount of power. However, nuclear power plants are being phased out in Sweden. At the same time, the consumption of electricity will increase by 46-60 % in the coming 25 years. The increase is primarily driven by the industry requiring large amounts of electricity to produce steel, free from coal [7]. The combination of the phase out of plannable generation capacity while having a significant increase in

consumption of electricity creates a unique challenge for the Swedish power system.

B. Paper set-up

In section II the theory behind capacity markets will be presented. How a capacity market could look like in Sweden is given in section III. The results are presented in section IV. An analysis of the results is presented in section V followed by the conclusions in section VI.

II. Theory

A. Marginal Pricing

A power generation unit's costs can be divided into two parts, *fixed costs* and *variable costs*. The fixed costs include capital costs, annual maintenance and wage costs that are independent of the production [8]. This differs from variable costs that depend on the production, such as costs for fuel and operating maintenance as well as costs for start and stop of the power plant [8].

The *marginal cost* of a generator is the cost of producing one additional unit of energy. This mainly corresponds to the variable cost, but for generators with a limited energy input, the *opportunity cost* adds to the variable cost [8]. This opportunity cost reflects the possible value of storing the energy and dispatch it in the future. For example, if a hydro-power plant with a limited amount of water stored in a dam forecasts a higher energy price in the future, the opportunity cost reflects missed income if the electricity is produced now rather than in the future.

In the short run, when a power plant is already built and ready to produce power, the marginal cost determines whether the power plant will be run or not. Selling energy to a price lower than the marginal cost would be a direct loss, while a price higher than the variable cost gives revenue. Therefore, in a fully competitive market, the price on an electricity spot market reflects the marginal cost of the most expensive unit required to meet the demand [8]. An example of this pricing model can be seen in figure 1.

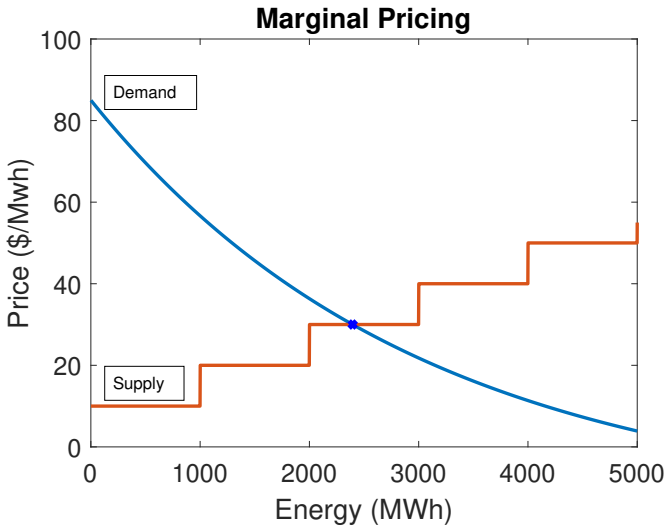


Fig. 1. An example of pricing in a competitive energy market for one hour. The blue line is the demand curve. The supply side consists of 5 power plants à 1000 MW with different marginal costs. The clearing price is decided by the most expensive plant needed to serve the demand.

B. Missing money problem

A market where producers get paid only for the energy they deliver is called an *energy-only market*. In a such market, the price of energy is set by the marginal cost of the most expensive unit needed to meet the demand, as seen in section II-A. Consequently, producers need to sell energy at a clearing price higher than their marginal cost for a certain time to cover their fixed costs [9]. As illustrated in figure 2, market distortions like price caps, can prevent prices from rising high enough to enable some power plants to cover their fixed costs [9]. This *missing money problem* can lead to insufficient investment in plannable energy sources and consequently create a resource adequacy problem [9].

Even without market distortions, a missing money problem can arise in an energy-only market. The market-based adequacy level can be lower than what policy makers and system operators consider acceptable [4]. Investing in new generation capacity is not profitable but may be necessary to solve the resource adequacy problem.

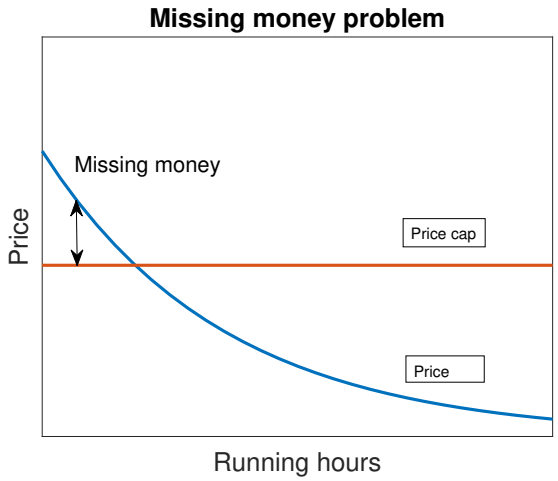


Fig. 2. A figure of the hourly price during a period, sorted from high to low. This price duration curve illustrates how the missing money problem can arise.

C. Capacity market

One way to improve the resource adequacy in a power system is to set up a *capacity market*. In this market, generators are paid for the capacity they can deliver to the power system, see figure 3. In a properly designed capacity market these payments solve the missing money problem by making it profitable to build and operate generators that maintain the reliability of the power system [4].

A capacity market can be designed in different ways. One common design in US is to oblige the load serving entities, LSEs, to secure enough power to meet their customers' peak load plus a certain reserve margin [4]. In a simple design, the LSEs sign agreements directly with power producers and thereby secure enough capacity. This decentralized market is called a *bilateral market* and is used in for example California [4]. These markets can suffer from problems with

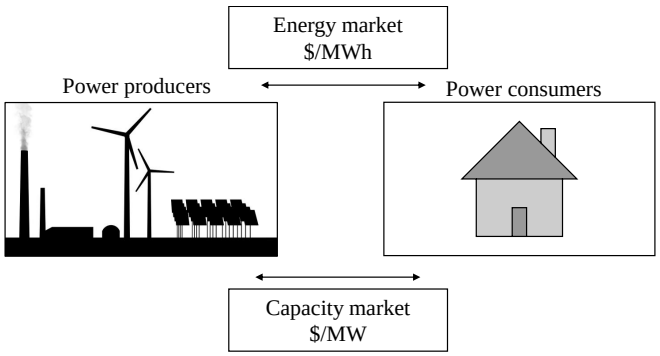


Fig. 3. The concept of an energy market with an additional market for capacity

price and volume transparency [4]. Another market design is a *mandatory centralized auction* where LSEs and producers participate. This kind of market is often combined with a bilateral market with a deadline [4]. If the LSE has not fulfilled their requirement on the bilateral market before a certain date, the remaining capacity requirement is cleared through the centralized auction. This kind of capacity market is used in for example New England and Pennsylvania. The time between the centralized auction and the delivery date on a capacity market is called the *forward period*.

D. Strategic reserve

The kind of capacity market mentioned in the previous section is primarily used in the US. In EU, it is more common to use a *strategic reserve* instead [10]. In a such market design, the buyer of capacity is the System Operator, SO, instead of the LSEs. The system operator sign agreements with power producers to secure enough capacity in power shortage situations. This market design is currently active in Sweden where Svenska Kraftnät can start an oil power plant in Karlshamn in shortage situations [11].

One big difference between a strategic reserve and earlier mentioned capacity markets is the number of producers getting capacity payments. Power plants in the strategic reserve receive an annual payment to cover fixed costs but are not allowed to participate in the energy market [11]. Primarily, power plants with high marginal cost and few operating hours will attend to the strategic reserve [10]. Power plants with low marginal cost will most likely earn more money in the energy market and will not participate in the strategic reserve [10].

The SO decides when the reserve plants are available in the energy market. The bid price of these plants in the energy market is placed higher than the most expensive competitive bid [11], which means that they will only be activated when all other power plants are insufficient. This pricing model lowers the risk of market distortion as the energy market is almost fully competitive [10]. Only when the market fails to deliver enough capacity, the strategic reserve intervenes.

In a capacity market the availability, or *capacity factor*, of different energy sources must be considered in the auction. It can be hard to determine that factor correctly for intermittent sources and flexible production, and this can lead to market distortion [10]. The capacity factor can be decided in different ways. In PJM for example, statistical analyses of availability in past years decide how much power a wind turbine is allowed to sell [12].

Because a strategic reserve does not have a holistic approach of the installed capacity, the system operators gain less control over the total installed capacity. In a capacity market it is easier to define a reliability target and guarantee enough power in the system to reach that target [10]. By choosing a long forward period, the capacity market can beyond solving the missing money problem, create long-term perspective and predictability in the market [4].

E. Value of Lost Load

Most consumers do not actively participate in the market for electricity. This demand side in-elasticity creates less flexibility in the market. Moreover, during times when supply cannot meet the demand, the physical properties of the power system result in the price setting not working. System flaws like these require a system operator to decide how much the demand side is willing to pay for reliability. The amount consumers are willing to pay to avoid a blackout is called the value of lost load (VOLL) [3]. VOLL is hard to estimate and can be derived in different ways. In Sweden, VOLL is estimated by directly asking consumers how much they value a loss of load. Using the cost of new entry (CONE) and dividing it by VOLL should, according to the following EU directive: [13], decides the reliability target. Other markets, like ISO-NE derives VOLL by deciding the reliability target they want. There is a key difference between the meaning of VOLL in NE and in the EU. It is also important to point out that VOLL varies greatly between consumers but in this paper the average VOLL estimate will be used as in [5].

F. Cost of New Entry

The Cost of new entry (CONE) is the annualized investment cost required to build and have new generation available. An explanation of how CONE is defined follows. Annualized capital expenditure (CAPEX) is the sum of the weighted average capital cost (WACC) and the annualized capital expenditure. Annual fixed costs consist of salaries, maintenance and costs associated with the building. The sum of these two annualized costs is equal to the CONE [14]. Net-CONE reflects whether it is profitable to construct new generation or not. It considers the earnings from that resource by subtracting it from the CONE. A positive net-CONE signals that the resource is unprofitable, that it has some missing money. In this project gas turbines will be the reference technology used to calculate CONE. Gas turbines are the most widely used reference technology because the generation is plannable. And they can quickly be built close to where they are needed [15].

G. Demand curves

As previously explained in the introduction, the demand side in the energy market does not participate actively. In order to clear a capacity auction, the amount of reliability the demand side is willing to pay must be represented in some way. This

is done through demand curves. Different capacity markets have constructed their demand curves in different ways. Most early markets had a fixed payment up to the installed capacity requirement (ICR) and some had a downwards sloping demand curve. Having a downward slope avoids a high price volatility and more markets are adopting variations of downward sloping curves. The max price for demand curves is set as a multiple of CONE or net-CONE to compensate for the missing money problem. A flatter slope lowers the price volatility while also increasing the risk of supply surplus and shortage as the price signal becomes weaker. An ideal demand curve is downwards sloping and convex which results in strong price signals while still having lower volatility than vertical curves [4].

III. Implementation

The model used in the simulations is retrieved from [5] and consists of a system with three different zones, see figure 4. The three zones are: the import constrained zone (ICZ), the export constrained zone (ECZ) and rest of the system (ROS). Together, these three regions form the total simulated system (SYS). Each zone has its own demand curve. The two zonal demand curves ICZ and ECZ represent the additional willingness to pay for capacity. In the import constrained zone, consumers benefit more from capacity being built within their zone than outside of it. The curve for the export constrained zone is negative, as more capacity in a zone where network constraints impeding exports does not impact reliability positively [5].

The capacity market uses a *Marginal Reliability Impact*-based demand curve, and the market process includes a Social Welfare Maximization-problem. This model was chosen because it used a convex downward sloping demand curve and such curves have many advantages, as described in section II-G.

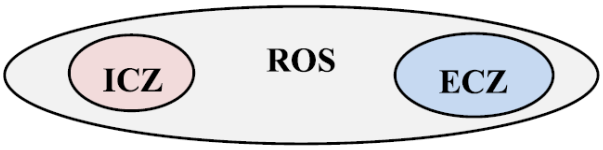


Fig. 4. The three system zones used in the optimization model. Source: [5]

A. Variable names

Q_{SYS}	Cleared demand in the whole system (MW)
Q_{ICZ}	Cleared demand in the ICZ (MW)
Q_{ECZ}	Cleared demand in the ECZ (MW)
$D_{SYS}(Q)$	Demand-curve function for whole system ($/kW)
$D_{ICZ}(Q)$	Demand-curve function for ICZ ($/kW)
$D_{ECZ}(Q)$	Demand-curve function for ECZ ($/kW)
q_{ROS}	Cleared capacity in ROS (MW)
q_{ICZ}	Cleared capacity in ICZ (MW)
q_{ECZ}	Cleared capacity in ECZ (MW)
$C_{ROS}(q)$	Cost of cleared capacity in ROS ($)
$C_{ICZ}(q)$	Cost of cleared capacity in ICZ ($)
$C_{ECZ}(q)$	Cost of cleared capacity in ECZ ($)
X	$= \{q_{ROS}, q_{ICZ}, q_{ECZ}, Q_{SYS}, Q_{ICZ}, Q_{ECZ}\}$
Z_q	$= \{ROS, ICZ, ECZ\}$
Z_Q	$= \{SYS, ICZ, ECZ\}$

B. Social Welfare Maximization

In order to maximize the social surplus, the market should minimize the cost of cleared capacity relative to the benefit for the society represented by the demand curves. This gives the following optimization model:

$$\begin{aligned} \min_{x \in X} \quad & \sum_{i \in Z_q} C_i(q_i) - \sum_{i \in Z_Q} \int_0^{Q_i} D_i(Q) dQ \\ \text{s.t.} \quad & q_{ROS} + q_{ICZ} + q_{ECZ} \geq Q_{SYS} \\ & q_{ICZ} \geq Q_{ICZ} \\ & q_{ECZ} \leq Q_{ECZ}. \end{aligned} \tag{1}$$

The first constraint says that the cleared capacity in the whole system must meet the total demand. Because the ICZ has limited import capacity, the cleared capacity must meet the demand within the zone. This gives the second constraint. In the ECZ, the export capability is limited and that gives the third constraint, setting an upper limit for cleared capacity in the ECZ.

Given the optimal solution for the demand Q^*_{SYS}, Q^*_{ICZ} and Q^*_{ECZ}, the capacity clearing prices are given by $D_{SYS}(Q^*_{SYS})$, $D_{ICZ}(Q^*_{ICZ})$ and $D_{ECZ}(Q^*_{ECZ})$ respectively [5].

C. Marginal Reliability Impact

The Marginal Reliability Impact (MRI) curve is the name of the approach used in ISO-NE [5]. The MRI curve for every FCM auction is publicly available and retrievable from [16]. In the simulations run in this project, data from the 2020/21 commitment period was used. As the name suggests, the additional impact on reliability is what the MRI curve shows. Impact on reliability is in this case the estimated unserved energy (EUE). An increase in the amount of unserved energy simply represents worse reliability in the system. Through the simulation software GE-MARS the RTO estimates the (EUE) and the corresponding derivative is evaluated at a range of capacity levels [5]. In order to reflect the value consumers put on a certain level of reliability, i.e. the demand, demand curves in the different zones are defined as follows:

$$D_{SYS}(Q_{SYS}) = -VOLL * \frac{dEUE_{SYS}(Q_{SYS})}{dQ_{SYS}} \tag{2}$$

$$D_{ICZ}(Q_{ICZ}) = -VOLL * \frac{dEUE_{ICZ}(Q_{ICZ})}{dQ_{ICZ}} \tag{3}$$

$$D_{ECZ}(Q_{ECZ}) = -VOLL * \frac{dEUE_{ECZ}(Q_{ECZ})}{dQ_{ECZ}} \tag{4}$$

D. Cost function

An arbitrary cost function was created in order to simulate the supply side of the market. In a real market, the cost function would represent bids from power producers. The cost function was constructed using general micro-economic assumptions from [17], stating a positive derivative and a positive second derivative. A positive derivative means that more cleared capacity leads to a higher price, i.e., the marginal price on capacity is positive. A positive second derivative means that the marginal price is increasing as the cleared capacity increases. This is logical because bids from cheap power plants are cleared in the first place and then more expensive ones. Another assumption is that the cost function should cross the origin since the cost of clearing no capacity is zero.

The cost function is represented using a quadratic function $C(q) = Aq^2 + Bq + C$. This yields the following system:

$$\begin{cases} 2Aq + B > 0 \\ 2A > 0 \\ C = 0 \end{cases} \tag{5}$$

The price of the cheapest capacity bid, is given by the derivative at $C'(0) = B$. Assuming a marginal price C_1' at an arbitrary point $q = q_1$, solving for A gives $A = (C_1' - B)/(2q_1)$. By adjusting the three parameters B, q_1 and C_1', different cost-functions can be created. For the simulation model of NE, three different cost-functions were created, one per zone. The parameters used in the simulation of NE are listed in table I and the C_{ROS} is plotted in figure 5. The corresponding parameters used in the simulation of Sweden are listed in table II.

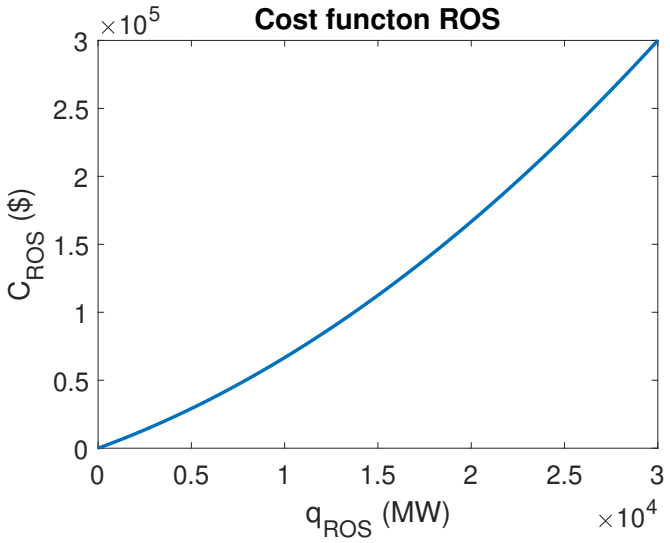


Fig. 5. The arbitrary cost function for the ROS zone in NE, defined by table I.

E. MATLAB implementation

The optimal solution of (1), given the ISO-NE demand curves and the cost functions, was found using the built in MATLAB-function *fmincon*. This solving method was chosen because the optimization problem is a *constrained nonlinear multivariable function* and *fmincon* can solve such problems. To use the MATLAB-function, the problem was reformulated into the optimization vector

$$x = \begin{bmatrix} q_{ICZ} & q_{ECZ} & q_{ROS} & Q_{SYS} & Q_{ICZ} & Q_{ECZ} \end{bmatrix} \tag{6}$$

and the matrix

$$A = \begin{bmatrix} -1 & -1 & -1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} \tag{7}$$

defining the constraint $Ax \leq \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$. The integrals in the optimization function were calculated using the function *trapz*. The whole MATLAB-script is available in appendix A.

All simulations were run at a Windows 10 64-bit computer with 16 GB RAM and an Intel Core i7-8550U CPU. The MATLAB version used was R2019a Update 8. The optimization workload was not very heavy, all MATLAB scripts had less than one second runtime.

TABLE I
PARAMETERS FOR COST-FUNCTION NE

Zone	A	C_1'	q_1
ICZ	5	15	10000
ROS	5	15	30000
ECZ	5	10	15000

TABLE II
PARAMETERS FOR COST-FUNCTION SWEDEN

Zone	A	C_1'	q_1
ICZ	5	10	11700
ROS	5	10	13200

F. Adaption for Sweden

As mentioned in the introduction, the Swedish power system will go through a significant transformation in the coming decades. Therefore, examining how a capacity market would perform in Sweden is interesting. To calculate an MRI demand curve for Sweden, a probabilistic reliability analysis like the one in ISO-NE is required. For ISO-NE, this analysis was done by a simulation software, GE-MARS, which was not available in this project. Probabilistic analyses of the Swedish power system have been published, but not in the desired format. For example, the system operator in Sweden, Svenska Kraftnät (SvK), has published a probabilistic analysis of the power system [18]. This report contains information about EUE and LOLE in the current system but does not contain any information about how the reliability would be affected by changes in available capacity. The LOLE value for Sweden during 2021 was 0.2 h/year [18].

In the absence of MRI data for the Swedish power system, an approximated SYS demand curve for Sweden was produced using the ISO-NE curve combined with some assumptions and data from SvK. The first assumption was that the MRI curve for Sweden would have the same shape as New England's curve. The shape of the MRI curve is decided by a combination of network constraints, locations of power plants and energy

mix. The second assumption made was that the EUE derivative and LOLE describes the same thing, as described by [13], giving the equation

$$\frac{dEUE}{dQ} = -LOLE. \tag{8}$$

Finally, the LOLE value for Sweden was assumed to correspond a capacity demand equal to the available capacity during the *peak load hour*, 24900 MW retrieved from [19]. The MRI curve was scaled with VOLL = 82.52 SEK/kWh calculated by [20].

In the south of Sweden (grid areas SE3 and SE4) there is more consumption than production during peak load hours and import of power is required to serve the demand [19]. The transmission capacity from north of Sweden is limited and there is often a price difference in the energy spot market between south and north of Sweden [19]. Under these facts, the south of Sweden was assumed to form an ICZ in the simulation. The same method as in the previous paragraph was applied to construct an ICZ demand curve. According to [18], all the loss of load is expected to occur in the south of Sweden, giving $LOLE = 0.2$ h/year. The available capacity during the *peak load hour* is 12700 MW [19]. The remaining two areas of Sweden, SE1 and SE2, were assumed to form the rest of the system, ROS. Because the LOLE in these areas is zero [18], there were no numbers available to construct an ECZ curve using the same method as for the whole system and the ICZ. Therefore, a two-zoned model was used in the simulation of Sweden.

G. Net CONE in Sweden

Examining the missing money problem in Sweden starts with calculating the net-CONE. Reference [21] provides data on the cost of building new gas turbines which is used to calculate CONE. Calculating net-CONE is done by examining the hours during the year the variable cost is exceeded by the spot price on electricity. Those hours the plant is profitable are summed up and subtracted from CONE resulting in the net-CONE. The spot prices used in the calculation were the market prices for Sweden (SE3) during 2020, downloaded from [22].

IV. Simulation results

A. Net-CONE

The CONE value for gas fired turbines in Sweden is 425 SEK/kW-year using the values from [21]. Subtracting the theoretical earnings from CONE results in a net-CONE of 375 SEK/kW-year. Sweden has by that result a missing money problem.

B. Cleared capacity for NE

Using the model described in section III, the optimal amount of cleared capacity in the respective zones can be seen in table III. Figures 6, 7 and 8 show the demand curves used in the simulation together with the optimal solution. As can be seen in 7 and 8, the market paid out more money to generation located in the import constrained zone and less to the export constrained zone. The total payment received is the cleared system price added by the zonal payment.

TABLE III
RESULT FROM SIMULATION OF NE

Zone	Q_i (MW)	q_i (MW)	Clearing price ($)
ICZ	9314.9	9314.9	14.315
ROS	15617	15617	10.206
ECZ	9310.5	9310.5	8.1035

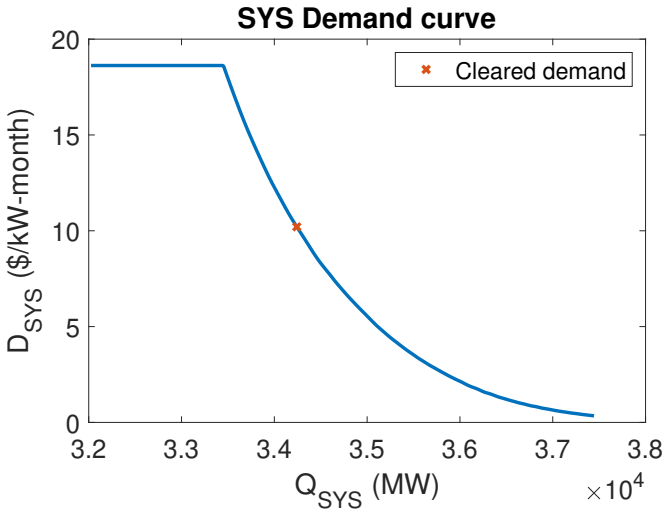


Fig. 6. Demand curve for SYS in ISO-NE. The demand curve data is adapted from [5]. The result from the optimization model using the cost-function from table I is marked.

C. Demand curve for Sweden

Figures 10 and 9 shows the resulting demand curves for SYS and ICZ in Sweden using the method described in III-F. The available capacity during the peak hour used in the derivation of each curve is marked. Table IV shows the optimal solution of system (1) using the two demand curves together with the cost function described in table II.

TABLE IV
RESULT FROM SIMULATION OF SWEDEN

Zone	Q_i (MW)	q_i (MW)	Clearing price (SEK)
ICZ	12779	12779	10.449
ROS	10246	10246	8.8873

V. Discussion

A. Net-CONE

The positive values of CONE and net-CONE could signal that there is some type of policy related issue that skews the market. However, the LOLE in Sweden is currently close to zero, which could mean that there is no missing money problem, and a positive net-CONE is just a sign of excess supply. Additionally, net-CONE was difficult to determine, and several factors contributed to this uncertainty. Firstly, the data retrieved from [21] was created seven years ago. The trend for gas fired power plants in the years following the report has been the construction of larger turbines lowering the per kW cost [15]. Therefore, CONE may not be accurate. Secondly, in the calculations of the earnings it was assumed that the power plant would be able to run every hour that

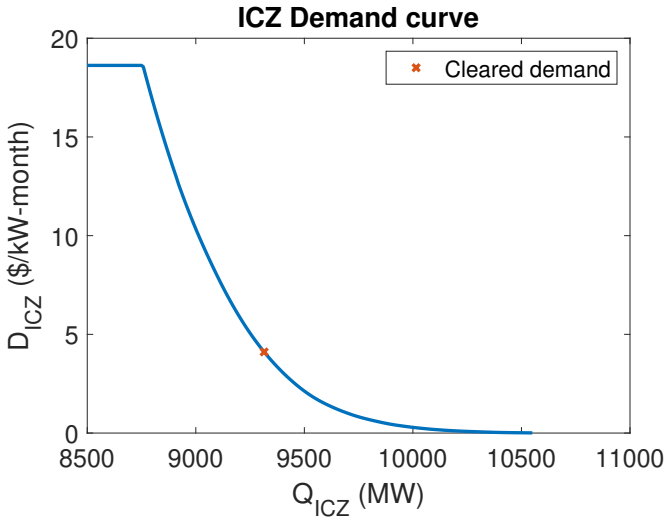


Fig. 7. Demand curve for ICZ in ISO-NE. The demand curve data is adapted from [5]. The result from the optimization model using the cost-function from table I is marked.

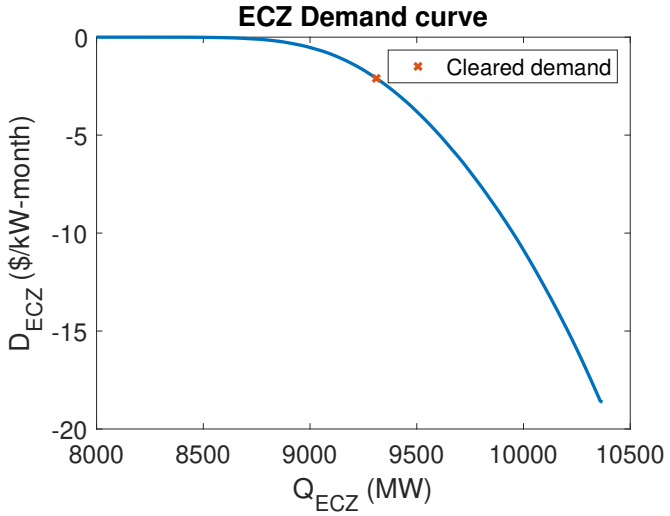


Fig. 8. Demand curve for ECZ in in ISO-NE. The demand curve data is adapted from [5]. The result from the optimization model using the cost-function from table I is marked.

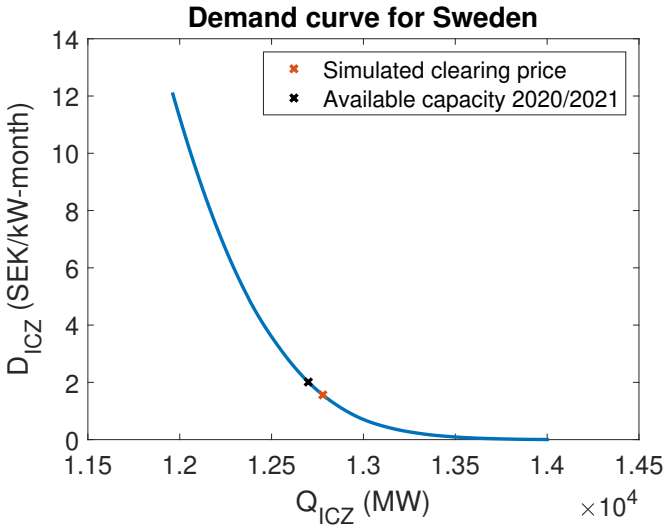


Fig. 9. Approximated demand curve for ICZ in Sweden using the method mentioned in III-F. The available capacity during peak hour of the winter 2020/2021 according to [19] is marked

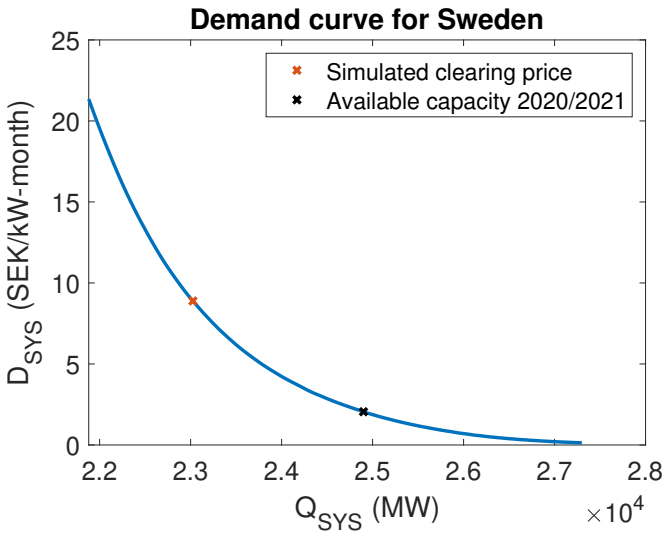


Fig. 10. Approximated demand curve for SYS in Sweden using the method mentioned in III-F. The available capacity during peak hour of the winter 2020/2021 according to [19] is marked

the spot price exceeded the variable cost. This assumption contributes to Net-CONE being lower than what is realistic as the earnings may be overestimated. The net-CONE in ISO-NE and PJM is calculated to 11.64 $/kW-month and 8.53 $/kW-month. Conversion to SEK and yearly prices gives a net-CONE of 1173 SEK/kW-year in ISO-NE and 860 SEK/kW-year in PJM [15] [5]. Even though currency fluctuations have changed since the Swedish net-CONE data were calculated, the difference compared to the American net-CONE values is sufficiently big to conclude that the missing money problem is more significant in those markets.

B. Clearing prices in Sweden

Despite using reliability results from ISO-NE, the adapted system curve for Sweden is still interesting. The available capacity during the peak load hour is the estimated reliable capacity that is available. This amount and mix of capacity are very close to the capacity potentially being able to participate in an FCM. Under those circumstances, assuming all reliable capacity participates, the cleared price would be extremely low compared to NE if a market where to be implemented in Sweden.

Another factor affecting the adaptation to Sweden is the different ways VOLL is calculated. As described in section II-E the key difference in determining the VOLL in the EU and the U.S. is whether the market or the SO decides the reliability target. A capacity market in Sweden would be more market-based than one in the U.S. because of this. Having less regulatory control, generally leads to a more effective market. In the example above, letting the SO decide the reliability target partly defies the purpose of implementing a capacity market. An FCM is supposed to compensate for

missing money caused by any regulations. Deriving VOLL as done in the U.S., may result in unnecessary high payments to power producers. The commonly used 1-in-10 criterion used to set reliability targets is an arbitrary value that only specifies the frequency of blackouts (1 every 10 years) and not the severity. On the one hand, administratively set targets can prove sub-optimal in a social welfare perspective. In other words, consumers may end up paying for more reliability than they want. On the other hand, surveys may not correctly represent the value consumers put on reliability, and therefore cause incorrect capacity payments.

C. Comparison with current strategic reserve

A FCM has to decide what share of the installed capacity from wind and other intermittent energy sources is allowed to participate in the market. Assuming the reliability of different power sources during peak hours described by [19], this mix would consist of more than 14000 MW of wind power and hydropower plants in Sweden. These power plants probably do not suffer from a missing money problem and may not need capacity payments. At the same time, the simulated capacity payments around 10 SEK/kW-month do not solve the missing money for gas turbines.

A strategic reserve has the advantage of only paying plants with potential profitability problems. Power plants in the strategic reserve do not participate in the normal energy market but is dispatched in shortage situation. The market participants will get payments to cover their fixed costs and primarily power plants with high marginal cost [10]. This kind of market would probably better address the missing money problem for gas turbines.

One advantage of an FCM however, is that a design with a forward period of several years will help avoid reliability issues in years with a high retirement of generation capacity. With a strategic reserve it is more difficult to control the total amount of installed capacity.

D. Long term perspective of resource adequacy

In the coming 20 years Sweden's power system will most likely see plannable generation like nuclear and cogeneration being phased out [19]. Together with increased demand, this could lead to a reduction in the reliability of the power system [23]. In the future, capacity mechanisms like FCM could be needed to ensure resource adequacy.

However, a sufficiently large strategic reserve can solve the resource adequacy as well. It could be an easier design because a capacity market would require different reliability indices, or capacity factors, to be calculated for different types of power plants. In a strategic reserve, only power plants with high marginal costs and few operating hours will participate [10] and the rest of the market will operate as usual. The challenge is to set the proper size of the strategic reserve. Today, Svenska Kraftnät has a multi-year contract with 562 MW of power valid until 2025 [11]. This may not be the most efficient way to handle a strategic reserve as the demand probably change throughout the years. An interesting solution would be to use a MRI-based demand curve to decide the size of the strategic reserve.

E. Future work

Simulating a capacity market in Sweden requires estimations on the reliability of the Swedish power system at all possible ranges where capacity could be cleared. Specifically, EUE values are needed to have something to base the demand curves on. These estimates can be done through Monte Carlo simulations. Other methods addressing the missing money problem, like better scarcity pricing through operating reserve demand curves, could contribute to solving the resource adequacy problem.

VI. Conclusion

To conclude, a forward capacity market in Sweden would under current circumstances result in payments well below net-CONE. Due to the fact that the Swedish power system has a high reliability in the current years, the missing money problem is not an immediate issue. Hence, the need for an FCM is not immediate but could be a useful tool in the future. In the close future, a strategic reserve is enough to ensure resource adequacy. Instead, constraints in the transmission grid is the more pressing matter.

Appendix A

MATLAB Script for the Optimization Model

Acknowledgment

We would like to express our gratitude to this project's supervisor, Mohammad Reza Hesamzadeh (Division of Electric Power and Energy Systems, KTH). His input, guidance and expertise were most helpful for the work done in this project.

References

[1] United Nations. (2021, Apr) The paris agreement. UNFCCC, Bonn, Germany. [Online]. Available: https://unfccc.int/process-and-meetings/the-paris-agreement/the-paris-agreement

[2] Hannah Ritchie and Max Roser. (2021, Apr) Co2 and greenhouse gas emissions. Global Change Data Lab, England. [Online]. Available: https://ourworldindata.org/co2-and-other-greenhouse-gas-emissions

[3] P. Cramton, A. Ockenfels, and S. Stoft, "Capacity market fundamentals," *Economics of Energy & Environmental Policy*, vol. 2, no. 2, pp. 27–46, 2013.

[4] K. Spees, S. A. Newell, and J. P. Pfeifenberger, "Capacity markets—lessons learned from the first decade," *Economics of Energy & Environmental Policy*, vol. 2, no. 2, pp. 1–26, 2013.

[5] F. Zhao, T. Zheng, and E. Litvinov, "Constructing demand curves in forward capacity market," *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 525–535, 2018.

[6] J. Bpwring, "Capacity markets in pjm," *Economics of Energy & Environmental Policy*, vol. 2, no. 2, pp. 47–64, 2013.

[7] A. Regnell, R.-M. Ågren, A. Wolf, E. Dotzauer, E. Mårtensson, F. Johnsson, G. Andrée, G. Melin, L. Flink, L. Hellman, S. Thorburn, and S. Larsson. (2019, Sep) Så klarar det svenska energisystemet klimatet. Stockholm, Sweden. [Online]. Available: https://www.iva.se/globalassets/bilder/projekt/vagval-klimat/201909-iva-vagval-for-klimatet-delrapport4-i.pdf

[8] D. R. Biggar and M. R. Hesamzadeh, *Efficient Short-Term Operation of an Electricity Industry with no Network Constraints*. John Wiley & Sons, Ltd, Chichester, UK, 2014, ch. 4, pp. 93–118. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118775745.ch04

[9] ——, *Market-Based Investment in Electricity Generation*. John Wiley & Sons, Ltd, Chichester, UK, 2014, ch. 10, pp. 199–208. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118775745.ch10

[10] Pär Holmberg and Thomas P. Tangerås. (2020, Oct) Incitamenten att investera i produktion på elmarknaden. Svenskt Näringsliv, Stockholm, Sweden. [Online]. Available: https://www.svensktnaringsliv.se/bilder_och_dokument/rapporter/incitamenten-att-investera-i-produktion-pa-elmarknaden_1151211.html/7edacdfd-a795-4ce1-895d-b52ea344fcc4.bin

[11] Svenska Kraftnät. (2020, Sep) Effektreserv. Svk, Sundbyberg, Sweden. [Online]. Available: https://www.svk.se/aktorsportalen/systemdrift-elmarknad/information-om-stodtjanster/effektreserv/

[12] L. Söder, E. Tómasson, A. Estanqueiro, D. Flynn, B.-M. Hodge, J. Kiviluoma, M. Korpås, E. Neau, A. Couto, D. Pudjianto, G. Strbac, D. Burke, T. Gómez, K. Das, N. A. Cutululis, D. Van Hertem, H. Höschle, J. Matevosyan, S. von Roon, E. M. Carlini, M. Caprabianca, and L. de Vries, "Review of wind generation within adequacy calculations and capacity markets for different power systems," *Renewable and Sustainable Energy Reviews*, vol. 119, p. 109540, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032119307488

[13] ENTSO-E. (2020, Apr) Proposal for a methodology for calculating the value of lost load, the cost of new entry for generation [...]. ENTSO-E AISBL, Brussels, Belgium. [Online]. Available: https://www.acer.europa.eu/en/Electricity/CLEAN_ENERGY_PACKAGE/Documents/Methodology%20for%20VoLL%20CONE%20and%20reliability%20standard_for%20submission%20to%20ACER.pdf

[14] K. Theodoropoulos, S. Daly, and M. Dinan. (2018, Sep) Cost of new entrant peaking plant and combined cycle plant. ENTSO-E AISBL, Brussels, Belgium. [Online]. Available: https://www.semcommittee.com/sites/semc/files/media-files/SEM-18-156a%20Poyry%20Report%20-%20Cost%20of%20New%20Entrant%20Peaking%20Plant%20and%20Combined%20Cycle%20Plant%20in%20I-SEM.pdf

[15] PJM. (2018, Apr) Pjm cost of new entry combustion turbines and combined-cycle plants. Brattle Group, Boston MA. [Online]. Available: https://www.pjm.com/~/media/committees-groups/committees/mic/20180425-special/20180425-pjm-2018-cost-of-new-entry-study.ashx

[16] ISO-NE. (2016, Sep) Fca-11 demand curve. ISO New England, Holyoke, Massachusetts, USA. [Online]. Available: https://www.iso-ne.com/static-assets/documents/2016/09/a2_fca11_demand_curve_revised.xlsx

[17] D. R. Biggar and M. R. Hesamzadeh, *Introduction to Micro-economics*. John Wiley & Sons, Ltd, Chichester, UK, 2014, ch. 1, pp. 3–30. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118775745.ch01

[18] Svenska Kraftnät. (2020, Dec) Kortsiktig marknadsanalys 2020. Svk, Sundbyberg. [Online]. Available: https://www.svk.se/siteassets/om-oss/rapporter/2020/kortsiktig-marknadsanalys-2020.pdf

[19] ——. (2020, Jun) Kraftbalansen på den svenska elmarknaden, rapport 2020. Svk, Sundbyberg, Sweden. [Online]. Available: https://www.svk.se/siteassets/om-oss/rapporter/2020/kraftbalansen-pa-den-svenska-elmarknaden-rapport-2020.pdf

[20] Maria Dalheim. (2021, Feb) Beräkning av värdet av förlorad last (voll). Energimarknadsinspektionen, Eskilstuna, Sweden. [Online]. Available: https://www.ei.se/download/18.6f9b6b2617714873b45f1838/1613489129164/Ber%C3%A4kning-av-v%C3%A4rdet-av-f%C3%B6rlorad-last-VoLL-Ei-PM2021-01.pdf

[21] Ingrid Nohlgren and Solvie Herstad Svärd and Marcus Jansson and Jennie Rodin. (2021, Apr) Electricity from new and future plants 2014. Energiforsk, Stockholm, Sweden. [Online]. Available: https://energiforskmedia.blob.core.windows.net/media/19920/electricity-from-new-and-future-plants-2014-elforskrapport-2014-45.pdf

[22] N. Pool. (2021, Jan) Historical market data - elspot prices in sek/mwh, 2020. Nord Pool AS, Lysaker, Norway. [Online]. Available: https://www.nordpoolgroup.com/48c964/globalassets/marketdata-excel-files/elspot-prices_2020_hourly_sek.xls

[23] Statnett, Fingrid, Energinet, and S. Kraftnät. (2018, Apr) Nordic perspectives on midterm adequacy forecast 2017. ENTSO-E AISBL, Brussels, Belgium. [Online]. Available: https://eepublicdownloads.entsoe.eu/clean-documents/SOC%20documents/Nordic/Nordic_perspectives_on_MAF_FINAL.pdf

Modellering av vattenkraft i Spine - En studie om gränserna för vattenkraften som reglerande energikälla i framtidens elsystem

Fredrik Lien Oscarsson and Tony Sibo

***Abstract*—In this paper a system of hydropower plants is analyzed with the aim of finding a breaking point at which the system can no longer handle an alternating electric load while keeping the water spillage at a minimum. A model of a part of Ångermanälven with ten power plants were implemented in the optimization software Spine in order to achieve this. The model was then supplied with data of two very challenging case scenarios during a year of operation. The result shows that the system, when met with a frequently changing load during the period of high local inflow, is able to sustain a high load while maintaining stability. On the other hand the case of having the lowest local inflow combined with the highest reservoir levels is also stable but only with a lower electric load. In conclusion; hydropower systems appear to play a key role in controlling potential variations in the power system due to its flexibility and storing capacity. A broader look at large hydropower systems in Sweden would, however, give us a clearer picture of how well these systems can serve as a compensatory energy source in future CO2-free power systems.**

***Sammanfattning*—I den här rapporten analyseras ett system av vattenkraftverk med målet att hitta brytpunkten för när systemet inte längre kan tillgodose en skiftande last samtidigt som spill av vatten hålls minimalt. En modell av en del av ångermannälven implementerades med tio kraftverk i optimeringsprogrammet Spine för att uppnå detta. Modellen kompletterades sedan med data från de två veckor då förhållandena för drift var som värst. Resultatet visar att systemet är stabilt när det belastas med en snabbt skiftande last och ger då högst elproduktion när tillrinningen är maximal. Veckan med lägst tillrinning samt högst vattennivå i reservoarerna modellerades också, dock med lägre elproduktion. Slutsatserna som kan dras är att system av vattenkraft verkar vara en viktig del i elproduktionen för att kunna kompensera potentiella variationer i framtida el-system. En mer omfattande studie och modell av vattenkraftverk skulle dock ge en bättre bild av hur väl dessa system kan agera som den kompensatoriska faktorn i framtidens koldioxidfria el-system.**

***Index Terms*—Hydropower, optimization, modelling, spillage, meeting electrical demand, future power systems.**

Supervisors: Mikael Amelin

TRITA number: TRITA-EECS-EX-2021:156

Symbol	Enhet	Definition
$\hat{H}_i$	MW	Maximal installerad effekt per kraftverk
$\mathrm{W}_{t,i}$	TE	Årlig medelvattenföring under timme t - vattenflödet i ett vattendrag
$\mathrm{L}_{t,i}$	TE	Lokal tillrinning till magasin i under timme t
$\hat{M}_i$	TE	Maximal kapacitet i magasin i
$\mathrm{M}_{i,start}$	TE	Vattenvolymen i magasin i vid startpunkten
$\mathrm{M}_{i,slut}$	TE	Vattenvolymen i magasin i vid slutpunkten
R_q	min	Rinntiden innan vattnet når nästa kraftverk

I. Introduktion

"Unless the sun dies, winds stop, plants die and rivers stop running, there will always be green energy to be had. Some of these energy sources are completely free and we have them no matter what. Why not take advantage of them?" [1]

Edgar Cervantes

En prisvärd, pålitlig och hållbar energikälla är nyckeln till hållbar utveckling. Med andra ord spelar energi en främjande roll för att stödja både social och ekonomisk välfärd genom att bidra till utrotningen av den globala fattigdomen, att säkerställa ett hälsosamt liv och höja levnadsstandarden i samhället. [2]

Å ena sidan, diskuteras ämnet om att de förnybara energikällorna i form av sol- och vindkraft är det sökta energislaget som behöver utökas för att kunna förse samhället med grön energi. Å andra sidan påpekas ofta svagheter i dessa typer av energi, till exempel deras tillförlitlighet när det gäller elproduktion under ogynnsamma väderförhållanden, såsom brist på vind eller antal soltimmar. Vattenfall, en stor aktör inom området för vattenkraftsproduktion, påstår att en planerad drift av vattenkraftverk i norden är en balanserande kraft som kan kopplas samman med sol- och vindkraft i andra europeiska länder. Vattenkraftverk utgör gigantiska batterier som lagrar elektricitet på ett indirekt sätt. Vatten kan nämligen lagras under perioder då elproduktionen från enbart vindkraftverk eller solceller tillgodoser efterfrågan och därmed ha ett energilager för att kunna styra elproduktionen med. På så sätt kan flexibiliteten hos vattenkraftsystem utnyttjas för att hantera extremt varierande laster eller en oregelbunden förnybar elproduktion, samt för att upprätthålla stabila frekvenser i elnätet [3].

En amerikansk studie inom ramen för vattenkraftsmodellering, utfört av det nationella laboratoriet för förnybar energi i USA, påpekar vikten av att förstå funktionerna och begränsningarna av flexibiliteten hos ett vattenkraftsystem för att kunna göra en effektivare nätplanering. Det vill säga, för att vattenkraft ska kunna drivas tillsammans med andra förnybara alternativ måste det först byggas kunskap om vilka utmaningar som försvårar en planerad drift av vattenkraftverk [4].

A. Syfte

Syftet med det här projektet är att undersöka flexibiliteten eller förmågan hos ett valt vattenkraftsystem att hantera extrema variationer i en fiktiv elektrisk last. Detta är för att få en insikt i de möjligheter som denna flexibilitet kan erbjuda för framtida sammankopplingar av vattenkraft och andra förnyelsebara alternativ. Till skillnad från tidigare studier kommer syftet

i detta arbete uppnås samtidig som vattenspillet från samtliga kraftverk i systemet ska hållas på en minimal nivå. Det vill säga, att hitta en optimal gräns där belastningen kan tillgodoses till det minsta möjliga vattenspillet.

B. Frågeställning

Hur kommer systemet, bestående av tio seriekopplade vattenkraftverk längs Ångermanälven, att bete sig när det utsätts för en elektrisk last med varierande effekt under bestämda tidsintervall?

C. Mål

- Framtagandet av en pålitlig modell som använder tillgängliga data för att hitta ett optimalt driftläge för det studerade systemet av vattenkraft.
- Att få kunskap om hur väl vattenkraft kan anpassa sig till extrema lastförändringar.
- Att presentera en fallstudie av en svensk älvsträcka som kan nyttjas för en fortsatt forskning inom området för planering och reglering av vattenkraftproduktion.

II. Bakgrund

A. Vattenkraft i Sverige

Användningen av vattenkraft för elproduktion i Sverige går tillbaka till 1880-talet. Ursprungligen användes el för belysning i större städer, men senare på 1890-talet började vissa industribolag bygga vattenkraftverk som skulle förse sina industrier med el. Följaktligen, har vattenkraft bidragit till ökad industrialisering i Sverige, vilket i sin tur har lett till ökad välfärd i samhället. Skogsindustrin, papper, stål och andra svenska industrier som använder den stabila och prisvärda elen från vattenkraftverk har utvecklats avsevärt under det senaste århundradet säger [5].

Idag finns drygt 2000 vattenkraftverk installerade på de svenska älvarna med en total effektkapacitet på cirka 16 TW. Den sammanlagda energin som produceras av samtliga vattenkraftverk under ett normalt år i Sverige är cirka 65 TWh el. Detta motsvarar omkring 30% högre produktion än vad som konsumeras årligen av hela den svenska industrin enligt [6]. Därmed utgör vattenkraften en av de största förnyelsebara energikällorna som utnyttjas för elproduktion i Sverige. År 2019 producerades drygt 165 MWh el, varav cirka 40% kom från svenska vattenkraftverk enligt [7].

B. Planering och balansering av elproduktion

Planeringen av elproduktionen från vattenkraftverk görs i form av korta, medellånga eller långsiktiga driftplaner. Beroende på vilken typ av planering som tillämpas, kan tidsintervallet för driftplaneringen sträcka sig från timvis till att vara på årsbasis baserat på prognoser för belastningen på elnätet och naturfenomen. Väder och klimat måste beaktas för att säkerställa att vattenvolymen i reservoarerna är tillräckligt under de planerade driftperioderna i enlighet med lagar och företagens intressen.

Eftersom elproduktionen konstant måste tillgodose efterfrågan på el, är det viktigt att känna till efterfrågan i förväg så att planeringen för framtidens elproduktion kan göras. För att minska risken för stora avvikelser mellan hur mycket elproduktion som har planerats och hur stor effekt som egentligen efterfrågas varje timme, kan frekvensreglering utnyttjas som ett mått för att matcha efterfrågan mer exakt med produktionen [8]. Denna process av att skapa balans mellan produktion och efterfrågan sköts dock till stor del av den nordiska elbörsen, Nord Pool, och dess interna marknader Elspot och Elbas. Varje aktör inom elmarknaden, Elspot, lägger bud på sin önskade volym av el givet i MWh/h för det kommande dygnet så att en 24-timmars elproduktion kan planeras ett dygn i förväg. För att kunna justera de redovisade elbehoven under leveransdagen, används en ytterligare balansmarknad som organiseras av Nord Pool, nämligen Elbas. Denna marknad tillåter aktörerna att justera sina behov av el fram till en timme innan leveranstimmen [9].

C. Optimering och modellering av vattenkraftproduktion

Optimeringslära definieras som teorin om att använda tillgängliga matematiska metoder eller modeller för att kunna maximera en vinst eller göra det bästa möjliga valet vid en beslutssituation [10]. Eftersom planeringen av vattenkraftproduktion ofta syftar till att optimera de ekonomiska och tekniska aspekterna av verksamheten, anses tillämpningen av olika optimeringsmodeller spela en väsentlig roll i planeringsprocessen. Beroende på vilka aspekter av vattenkraftproduktionen som vill optimeras kan modellerna byggas på diverse olika sätt och därmed ha olika målfunktioner. Det kan exempelvis handla om att maximera vinsterna från såld el, att driva kraftverken med maximal verkningsgrad genom att fördela vattenflöden mellan turbinerna på ett optimalt sätt och därmed generera högsta möjliga effekt, eller att försöka minimera vattenförlusterna tillsammans med andra bivillkor. Hur avancerad modellen kan göras, eller hur lik verkligheten simuleringarna kan bli, beror vanligtvis på vilka faktorer som inkluderas eller beaktas vid modelleringen. Enkla modeller konstrueras ofta som ett första steg sedan uppgraderas dessa genom att inkludera ytterligare parametrar och därmed kan mer rimliga resultat uppnås. Trots att modeller som representerar olika system av vattenkraftverk ofta har olika målfunktioner, finns det fortfarande en gemensam utgångspunkt för de flesta modellerna. Denna utgångspunkt är nämligen de grundläggande data som matas in i modellen [11].

I följande underavsnitt presenteras några huvudbegrepp relaterade till dessa data och som anses vara mest använda i vattenkraftsmodellering.

1) Tappning: Med tappning avses den mängd vatten som per tidsenhet leds till en eller flera turbiner som finns installerade vid kraftverket. Den rinnande vattenvolymen utnyttjas för att få turbinerna att rotera och därigenom omvandla den mekaniska energin till elektrisk energi. Data om tappning behövs vid modelleringen av vattenkraftproduktion oavsett vilket syfte modelleringen har, eftersom detta är en avgörande faktor när det gäller hur mycket el som kan genereras [11].

2) Spill: Med spill menas däremot vattenvolymen som måste spillas förbi kraftverket utan att vattnet leds genom turbinerna. Det vill säga, ingen elektricitet produceras och vattnet går förlorat. Detta kan exempelvis ske när ett vattenkraftverk drivs med full tappning medan ett nedströms liggande kraftverk är avstängt med fullt magasin. Vattnet som strömmar in i de nedersta magasinen kommer behöva spillas så att den hydrologiska balansen ska upprätthållas [12].

$$\begin{aligned}\text{Nytt innehåll i magasin} = {} & \text{gammalt innehåll i magasin} \\ & + \text{inflöde till magasin} \\ & - \text{utflöde från magasin}\end{aligned} \quad (1)$$

3) Tillrinning: Är i princip allt vatten som rinner in i reservoaren från direkt nederbörd såsom regn eller från snösmältning under vårsäsongen. Förutom att mängden nederbörd per år har en direkt påverkan på tillrinningen finns det ett flertal andra faktorer som avgör storleken på den årliga tillrinningen. Dessa är exempelvis arean av avrinningsområdet kring vattenmagasinen, vegetationen över de naturliga avrinningsytorna, typ av mark och potentiella avdunstningvolymer från anslutna sjöarealerna [11].

4) Magasin: Ett magasin avser reservoaren bakom dammen där tillrinningen samlas upp och lagras. Vattenflödet i vattendraget regleras sedan genom en kontrollerad tappning. Fyllnadsgraden i ett magasin bestäms av bland annat mängden tillrinning per år och ju högre fyllnadsgrad, i samband med hög tillrinning, desto större elproduktion kan planeras. En annan fördel med en hög fyllnadsgrad i magasinet är att vattenvolymen kan hållas hög och därmed skapa högre effekt per volymenhet vatten. Data om maximalt tillåten fyllnadsgrad i ett magasin eller fyllnadsgraden vid en viss tidpunkt är en huvudparameter som ingår i vattenkraftsmodellering och som bestäms av magasinets kapacitet. Däremot representeras den minimalt tillåtna fyllnadsgraden ofta av fastställda värden från vattendomarna. En vattendom avgör i vilken utsträckning tappningen av en älv får utföras [11].

5) Sparat vatten: I vilken utsträckning det lagrade vattnet i ett magasin kan utnyttjas är en fråga som besvaras av bland annat hur högt elpriset är under de planerade driftperioderna. Det är med andra ord viktigt att ta hänsyn till elpriserna vid planeringen av framtida drift av vattenkraftverk eftersom detta är an avgörande faktor som bestämmer lönsamheten av produktionen. Det kan exempelvis vara fördelaktigt att stoppa elproduktionen från ett visst vattenkraftverk när elpriset är extremt lågt och spara vattnet istället. Mängden lagrat vatten kan återanvändas vid ett senare tillfälle då elpriset har stigit [12].

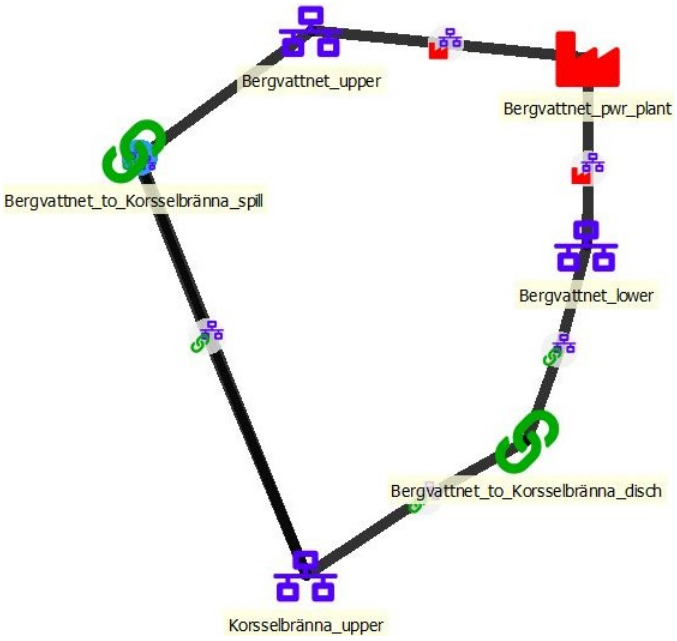
III. Fallstudie

Under följande avsnitt presenteras optimeringsprocessen för tio seriekopplade vattenkrftverk längs Ångermanälven, nämligen Dabbsjö, Bergvattnet, Korsselbränna, Tåsjö, Hoting, Borgforsen, Bodum, Fjällsjö, Sil och Kilforsen.

A. Modellering

För att optimera älvsträckan med de valda vattenkraftverken mot en påfrestande last med minimalt spill har modelleringsprogrammet Spine använts med ett tillägg som heter SpineOpt. Detta tillägg tillåter programmet att både modellera och simulera driften av samtliga vattenkraftverk.

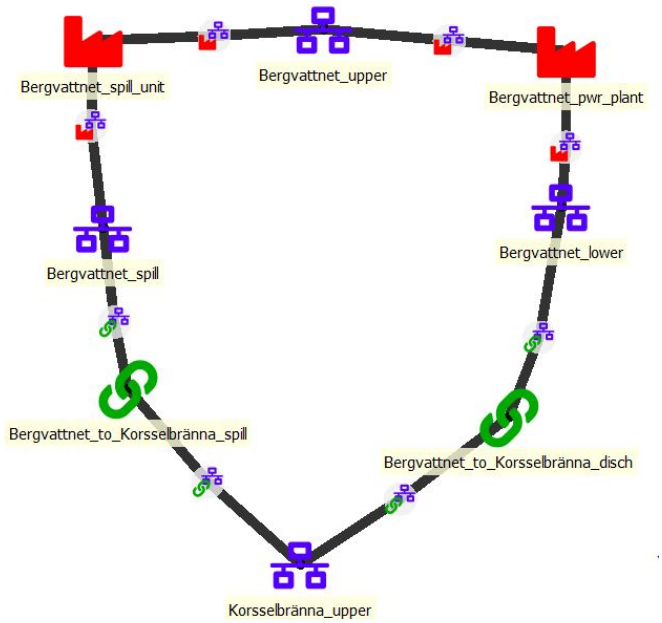
Modellen som byggdes för det här arbetet är baserad på en tidigare modell gjord i Spine som optimerade en älvsträcka på Skellefteälven mot maximal ekonomisk vinning. För att få fram modellen för Ångermanälven byggdes först en kopia av den tidigare modellen fast med de kraftverk och respektive data för Ångermanälven. Vid det tillfället fanns en modell för Ångermanälven som optimerade kraftverken för monetär vinning, det vill säga att modellen strävade efter att driva kraftverken med maximal verkningsgrad under de timmar då elpriset och elproduktionen var som högst.



Figur 1. Strukturen för ett av de modellerade kraftverken utan mätning av spill.

För att bygga om modellen för optimering mot minimalt spill lades det till ett objekt per kraftverk som kunde mäta mängden spill per kraftverk. Detta kan ses i den vänstra armen av figur 2 där objektet $Bergvattnet_spill_unit$" har lagts till för att mäta mängden spillvatten som flödar förbi kraftverket $Bergvattnet$". Utöver detta ändrades även modellens målfunktion genom att ställa in varje enhet spill till en fixerad imaginär kostnad. Detta så att modellen skulle jobba mot att leverera den begärda effekten samt undvika det kostsamma spillet. För att kunna tillgodose den elektriska lasten fastställdes den begärda mängden el per timme från samtliga kraftverk till det värdet som lastkurvan hade under varje timme. De objektändringar som gjordes för kraftverket $Bergvattnet$" i denna modell illustreras i figur 1 och figur 2 och resten av kraftverken har genomgått samma ändringar.

När modellen var klar påbörjades arbetet för att hitta brytpunkten för när systemet inte längre kunde tillhandahålla den



Figur 2. Strukturen för ett av de modellerade kraftverken med mätning av spill.

begärda energin. Utgångspunkten för detta var en varierande last formad som en fyrkantsvåg mellan 10% och 90% av den totala effekten som kraftverken kan producera. Sedan testades olika tidsintervall på denna last för att se när det blev svårt för modellen att leverera de önskade resultaten. Den initiala takgränsen för den begärda elproduktionen från modellen, nämligen 90%, sänktes successivt tills det att gränsfallet hittades.

B. Data

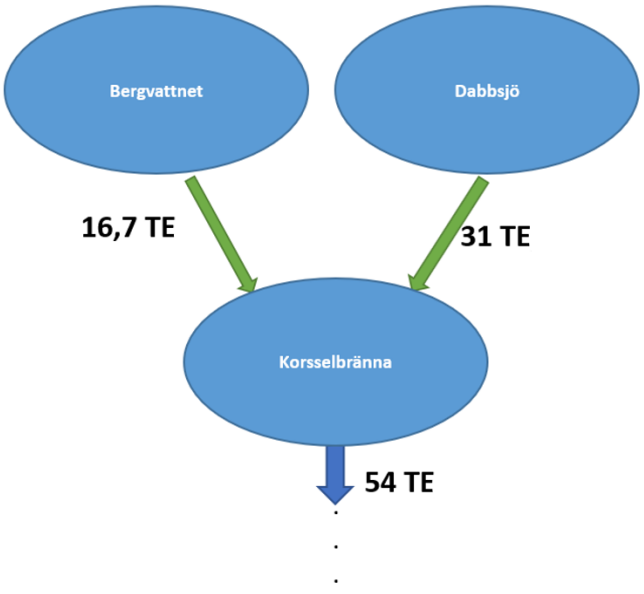
Data för maximal installerad effekt, årlig medelvattenföring och rinntiden för berörda kraftverk baseras på verkliga värden som har tillhandahållits av projekt beställaren. Dessa värden har jämförts med tillgängliga data från [13] för att kunna kontrollera dess noggrannhet samt säkerställa rimliga resultat. I tabell I presenteras dessa data för samtliga kraftverk. Rinntiden för tappningar är inte konstant i verkligheten utan den påverkas av ett antal faktorer såsom säsong, klimat och vattennivåerna i magasinen. Under vintertid till exempel fryser den största delen av ytvattnet i flodsystemen när temperaturerna har sjunkit tillräckligt och därmed tar det längre tid för vatten att flöda mot nästa station. För att förenkla problemet har rinntiden mellan två efterliggande kraftstationer antagits vara konstant vilket anses vara ett rimligt antagande enligt [12].

Den lokala tillrinningen för Dabbsjö och Bergvattnet i tabell III har fastställts till ett konstant värde som är lika med den årliga medelvattenföringen för respektive kraftverk. Detta är eftersom Dabbsjö och Bergvattnet ligger i början av den studerade älvsträckan och båda utgör ett uppströms flöde som mynnar ut i Koresselbrännas magasin. Från och med Korsselbränna har den lokala tillrinningen beräknats genom att subtrahera den årliga medelvattenföringen, som rinner in i kraftverkets magasin, från det som lämnar samma kraftverk.

Tabell I
INMATADE DATA FÖR MODELLERING AV VATTENKRAFTVERK [13]

Kraftverk	$\hat{H}_i$ (MW)	$W_{t,i}$ (TE)	R_q (min)
Dabbsjö	26	16,7	65
Bergvattnet	21	31	40
Korsselbränna	130	54	15
Tåsjö	13	62	60
Hoting	13	78	20
Borgforsen	26	116	120
Bodum	12	117	60
Fjällsjö	13	122	30
Sil	12	122	120
Kilforsen	288	131	20

Till exempel har tillrinningen för Korsselbränna under timme t beräknats enligt figur 3 och ekvation 2



Figur 3. Schematisk bild av de tre första kraftverken i systemet under timme t.

$$\begin{aligned} \text{Tillrinning} &= \text{Utgående flöde} - \text{Inkommande flöde} \\ &= 54 - (31 + 16,7) \\ &= 6,3\,TE \end{aligned} \tag{2}$$

Fyllnadsgraden i magasinen vid startpunkten av första simuleringen, det vill säga första timmen av vecka 16 april 2019, motsvarar 13,6% av den maximala magasinkapaciteten. Medan i slutet av simuleringen, nämligen sista timmen av vecka 16, ökar fyllnadsgraden till 23,2%. Se tabell II. Dessa procentenheter baseras på uppmätta data för magasinsfyllnaden i område SE2 för vecka 16 och 17, det vill säga samma område där samtliga vattenkraftverk är belägna [14]. I tabell III presenteras fyllnadsgraderna för respektive reservoar för start och slut vid den andra simuleringen. Fyllnadsgraden första timmen vecka 41 är 81,2% och minskar till 79,6% i slutet av

Tabell II
INMATADE DATA FÖR MODELLERING AV SAMTLIGA MAGASIN VECKA 16 [13]

Magasin	$\hat{M}_i$ (TE)	$M_{i,start}$ (TE)	$M_{i,slut}$ (TE)	$L_{t,i}$ (TE)
Dabbsjö	93 610	12 731	21 738	83,5
Bergvattnet	3 610	491	838	155
Korsselbränna	9 720	1 322	2 255	31,5
Tåsjö	72 780	9 898	16 885	40
Hoting	6 390	869	1 482	80
Borgforsen	1 500	204	348	190
Bodum	5 928	806	1 375	5
Fjällsjö	1 100	150	255	25
Sil	100	14	23	0
Kilforsen	1 950	265	452	45

Tabell III
INMATADE DATA FÖR MODELLERING AV SAMTLIGA MAGASIN VECKA 41 [13]

Magasin	$\hat{M}_i$ (TE)	$M_{i,start}$ (TE)	$M_{i,slut}$ (TE)	$L_{t,i}$ (TE)
Dabbsjö	93 610	76 011	74 514	16,7
Bergvattnet	3 610	2 931	2 874	31
Korsselbränna	9 720	7 893	7 737	6,3
Tåsjö	72 780	59 097	57 933	8
Hoting	6 390	5 189	5 086	16
Borgforsen	1 500	1 218	1 194	38
Bodum	5 928	4 814	4 719	1
Fjällsjö	1 100	893	876	5
Sil	100	81	80	0
Kilforsen	1 950	1 583	1 552	9

veckan. Även datan för det här fallet kommer från uppmätta värden i område SE2 [15].

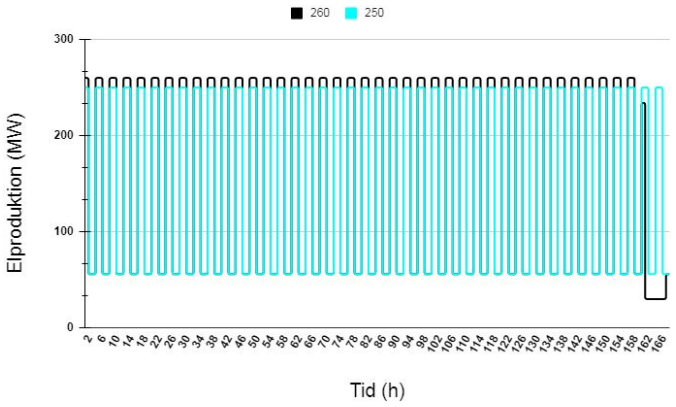
Anledningen till att vecka 16 och 41 har valts att undersökas är att dessa veckor hade den lägsta, respektive den högsta, fyllnadsgraden i magasinen år 2019.

C. Resultat

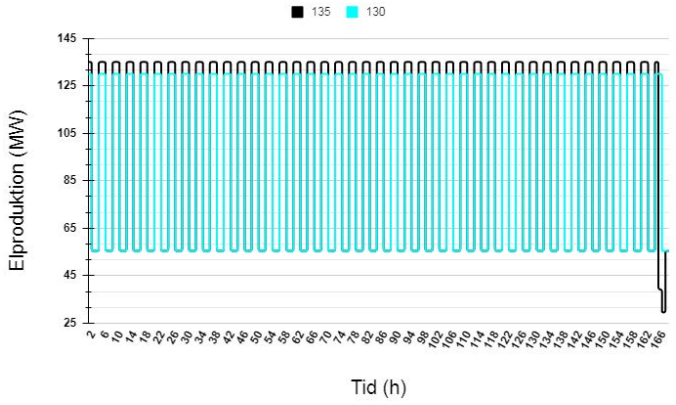
Två olika veckor modellerades. Vecka 16, 2019, och Vecka 41, 2019. Båda veckorna är modellerade med noll spill.

I figur 4 visas gränsfallet för när systemet inte längre kan tillgodose lasten med noll spill under denna vecka. Den turkosa kurvan är för en last mellan 55,4-250 MW med 2 timmars intervall, formad som en fyrkantsvåg. Den svarta kurvan är med samma värden på allt förutom lastens intervall, som i detta fall är mellan 55,4-260 MW. Figur 4 visar att systemet klarar den förstnämnda lasten men inte den andra, båda med noll spill. I det andra fallet, det vill säga den svarta linjen, kan systemet inte uppehålla den elproduktion som efterfrågas. Därav avviker den svarta linjen från formen av en fyrkantsvåg. För dessa optimeringar användes en skalfaktor på den lokala medeltillrinningen för varje kraftverk med syfte till att ha tillräckligt med vatten i systemet och därmed kunna simulera högre laster. Denna skalfaktor var fem.

Figur 5 visar, likt figur 4, gränsfallet för drift. Men i detta fall för vecka 41. I denna figur är den turkosa kurvan en last mellan 55,4-130 MW. Den svarta kurvan är för intervallet 55,4-135 MW. Från denna figur framgår det att systemet klarar den lägre lasten men inte den högre. Båda med noll spill. Det vill säga att körningen som representeras av den svarta kurvan inte kunde tillgodose lasten. Ingen skalfaktor användes för medeltillrinningen i detta fall. Figur 4 och figur 5 är bifogade i full storlek under bilaga 6.



Figur 4. Modellering av vecka 16



Figur 5. Modellering av vecka 41

D. Diskussion

När förnyelsebara energikällor talas om är det främst sol- vind- och vattenkraft som nämns. Dessa är tre välanvända förnyelsebara alternativ när det gäller elproduktion och således bland de mest relevanta. Om bara några år har Sverige och många andra länder målet att helt avveckla fossila energikällor samt kärnkraft. För att uppnå detta kommer en stor del av världens elproduktion bestå av de tre tidigare nämnda förnyelsebara energikällorna. Vattenkraft är ett av alternativen som kan lagra energi och kommer därmed behöva vara en av de reglerande faktorerna i framtidens elsystem. De modelleringar som genomförts i detta arbete är scenarion som ska spegla de värsta tänkbara förhållandena för vattenkraften i framtidens elproduktion. Där solenergi och vindkraft med flera förnyelsebara energi alternativ genererar mycket el under två timmar och sedan lite el under två timmar. Det är under dessa timmar då elproduktionen, från allt annat än vattenkraften, är låg som denna måste vara den energikälla

som kompenserar för detta. Det är dock viktigt att inte slösa på potentiell energi. Därav ska spillet från kraftverken hållas minimalt. De simuleringar som gjorts i det här arbetet tyder på att vattenkraften är bra för att tillhandahålla rätt mängd el vid de tillfällen då den krävs. Det finns dock begränsningar på vilka laster som kan tillgås, speciellt om reservoarerna inte får tömmas för mycket vid ett givet tillfälle. Vilket är ett rimligt antagande då det måste finnas lagrad energi kvar för andra tidpunkter då den behövs.

En viktig aspekt att ha i åtanke är att dessa resultat endast är baserade på en kort del av en älvsträcka. Hade modellen implementerats med alla vattenkraftverk på hela Ångermanälven skulle mer utförliga analyser kunna göras. Dessutom är en del data, såsom tillrinning och rinntider uppskattade. Den procentuella nivån, avseende mängden vatten, i alla reservoarer har även satts till samma procentsats av respektive reservoars maxkapacitet. I verkligheten är det högst otroligt att detta skulle inträffa. Således är modellen i detta arbete både en förenkling av verkligheten samt en liten del av vattenkraftsnätverket i Sverige. Trots detta anses denna modell vara tillräcklig för att få en bra uppfattning om hur väl vattenkraften kan komma att möta de krav som ställs på detta energialternativ i framtiden.

De två veckorna som modellerades valdes med tanken att utforska två extremfall. Vecka 16 är den vecka där magasinens fyllnadsgrad ökar som mest vilket betyder att även tillrinningen är som högst vid denna vecka. På grund av den drastiskt ökade tillrinningen har en skalfaktor på 5 gånger använts för medeltillriningen under året, vid optimeringen av denna vecka. Reservoar-volymerna är även som lägst, under hela året, vid början av vecka 16. Vecka 41 är det andra extremfallet det vill säga motsatsen till vecka 16. Denna vecka är reservoar nivåerna som högst vid början av tidsperioden och avtar endast med 1,6% till veckan efter. Tillrinningen denna vecka är nära medelvärdet av tillrinningen över hela året. Därav användes ingen skalfaktor för tillrinningen vid denna optimering. Valet av att använda datan från 2019 härstammar från att värdena för detta år var nära snittet på värdena över alla år sedan 1960.

Båda fallen optimerades med en nolltolerans för spill. Från resultaten framgår det att tillrinning såväl som reservoarnivåer sätter begränsningar på vilken last som systemet kan tillhandahålla. Under vecka 16 kan systemet möta en last mellan 55.4 till 250 MW. Vecka 41 klarar dock modellen endast att tillgodose en last mellan 55.4 till 130 MW trots att reservoarnivåerna är högre under denna vecka samt att de minskar under veckan. Anledningen till detta är att tillrinningen under vecka 16 är fem gånger högre än vid vecka 41. Reservoarerna fylls på markant under denna vecka men mer el kan ändå produceras än under vecka 41. Orsaken till att det blir så, är att mer vatten kan passera turbinerna under vecka 16. De reservoarer som ligger nedanför ett annat kraftverk, vilket är fallet för alla utom Dabbsjö och Bergvattnet, fylls till stor del på av vatten som redan passerat en eller flera turbiner och därmed redan genererat el. Den markant ökade tillrinningen under vecka 16 är den bidragande faktorn till att mer el kan produceras samtidigt som reservoarerna fylls på. Skalfaktorn på fem gånger den årliga medeltillrinningen är vald eftersom extremfall skulle studeras. En sådan stor tillrinning är ovanlig men kan förekomma.

Lasterna valdes som fyrkantsvågor med tidsintervallet två timmar eftersom en fyrkantsvåg är den typ av last som sätter störst press på ett system med vattenkraftverk. Tidsintervallet valdes efter tester med olika tidsintervall där det framgick att snabbare intervall gjorde det svårare för modellen att uppnå kraven. Det var dock ingen tydlig skillnad mellan svårighetsgraden i intervallen: en timme och två timmar. Två timmars intervall valdes för mer överskådliga resultat. Det är dock viktigt att poängtera att en sådan last antagligen aldrig skulle uppkomma i verkligheten. De verkliga lasterna är mycket mer komplicerade samt ej periodiska i största utsträckning.

Den nedre gränsen av fyrkantsvågen valdes som 55.4 MW då det är 10% av maximal installerad effekt från samtliga kraftverk. Den övre gränsen i båda fallen togs fram genom tester med ökad last till dess att systemet inte längre kunde leverera den begärda effekten. Där 250 MW är cirka 45% och 130 MW är cirka 23% av den maximala installerade effekten. När belastningen är något över vad systemet, under de då rådande förutsättningar, kan tillhandahålla visar resultaten något intressant. I båda fallen klarar systemet att uppfylla de krav som ställs ända fram till de sista timmarna. Detta beror på att det inte finns nog med vatten i systemet. Simuleringar gjordes även med högre laster samt med andra lastformer, så som triangelvåg. I alla dessa simuleringar hade systemet inget problem med att tillgodose lasten i början av simuleringen. Det var först vid dag två till dag tre som den begärda energin inte längre kunde levereras i de fallen. Bristen av energi förekom desto tidigare desto högre last som krävdes av systemet. Det var således inget problem för systemet att klara av ännu större last skiftningar, med noll spill, under en kortare tid men på grund av vattenbrist gick det då inte att tillgå lasten en hel vecka.

För att fullända denna modell och därmed göra den mer verklighetstrogen skulle den lokala tillrinningen för respektive vattenkraftverk behöva mätas mer exakt. Utöver detta borde modellen göras mer omfattande genom att inkludera alla vattenkraftverk på Ångermanälven. Om dessa ändringar i samband med en verklighetstrogen last implementeras skulle denna modell kunna användas för att planera driften för denna del av Sveriges vattenkraftproduktion.

IV. Slutsatser

Hur väl vattenkraft kan agera som reglerande energikälla i samband med minimalt spill av vatten har studerats i detta arbete. Studien genomfördes i form av en simulering i optimeringsprogrammet Spine. Resultaten studerades därefter för att dra följande slutsatser.

- Vattenkraft är en väldigt bra reglerande energikälla på grund av lagringsmöjligheterna som finns för denna.

- Vattenkraft kan möta mer krävande laster under vissa perioder på året än andra. Är som bäst när tillrinningen är som högst.
- Det förefaller högst möjligt att framtidens energibehov ska kunna tillgodoses av enbart förnyelsebara energialternativ, åtminstone i Sverige.
- För att förbättra denna studie behöver fler kraftverk involveras i modelleringen och datan som används måste vara mer exakt samt tidsanpassad.

BILAGA 1

Bilaga 1 är en förenklad bild över hur systemet är sammankopplat.

BILAGA 2

Bilaga 2 är en graf över hur mycket el vardera kraftverk producerar under vecka 16 med last mellan 55.4 till 250 MW.

BILAGA 3

Bilaga 3 visar spill av vatten per kraftverk vecka 16.

BILAGA 4

Bilaga 4 är en graf över hur mycket el vardera kraftverk producerar under vecka 41 med last mellan 55.4 till 130 MW.

BILAGA 5

Bilaga 5 visar spill av vatten per kraftverk vecka 41.

BILAGA 6

Bilaga 6 visar simuleringsgraferna för vecka 16 och vecka 41 i full storlek.

TILLKÄNNAGIVANDE

Vi vill tacka vår handledare Mikael Amelin för hans hjälp med projektet i sin helhet och den data han tillhandahållit. Manuel Marin ska också ha ett stort tack för all hjälp han erbjudit för Spine och arbetet med modelleringen.

REFERENSER

[1] E. Cervantes. (2016, Mars) 8 ways green energy is going to change the world. [Online]. Tillgänglig på: https://www.androidauthority.com/8-ways-green-energy-is-going-to-change-the-world-678167/

[2] UNECE, "Pathways to sustainable energy - accelerating energy transition in the unece region," pp. 1–2, April 2020. [Online]. Tillgänglig på: https://unece.org/fileadmin/DAM/energy/se/pdfs/CSE/Publications/Final_Report_PathwaysToSE.pdf

[3] Vattenfall. (2021, April) Vattenkraft. [Online]. Tillgänglig på: https://group.vattenfall.com/se/var-verksamhet/vara-energislag/vattenkraft

[4] B. Stoll, J. Andrade, S. Cohen, G. Brinkman, och C. B. Martinez-Anido, "Hydropower Modeling Challenges," National Renewable Energy Laboratory, U.S. Department of Energy, Denver West Parkway, Golden, Colorado, Tech. Rep., April 2017. [Online]. Tillgänglig på: https://www.nrel.gov/docs/fy17osti/68231.pdf

[5] J. Helbrink, J. Linnarsson, E. Hagner, M. Brolin, M. Löfqvist, och S. Nordquist, "Vattenkraft - påstående fakta," pp. 10–12, 2015. [Online]. Tillgänglig på: https://www.fortum.se/sites/default/files/documents/vattenkraft-pastaenden-och-fakta-20150701_1.pdf

[6] K. Lindholm. (2020, Nov) Vattenkraftsproduktion. [Online]. Tillgänglig på: https://www.energiforetagen.se/energifakta/elsystemet/produktion/vattenkraft/vattenkraftsproduktion/

[7] SCB. (2021, Feb) Elektricitet i sverige. Sverige i siffror. [Online]. Tillgänglig på: https://www.scb.se/hitta-statistik/sverige-i-siffror/miljo/elektricitet-i-sverige/

[8] O. Tengberg, "Implementation of hydro power plant - optimization for operation and production planning," Master's thesis, 2019. [Online]. Tillgänglig på: http://www.diva-portal.org/smash/get/diva2:1321561/FULLTEXT01.pdf

[9] Energimarknadsinspektionen. (2021, Mars) Elmarknader och elhandel. [Online]. Tillgänglig på: https://www.ei.se/sv/for-energikonsument/el/Elmarknader-och-elhandel

[10] J. Lundgren, *Optimization*, 1st ed. Lund, Sweden: Studentlitteratur AB, 2010.

[11] M. Sunnefors och T. Vainionpää, "Optimering av ett småskaligt vattenkraftsystem," Master's thesis, 2005. [Online]. Tillgänglig på: http://www.diva-portal.org/smash/get/diva2:604515/FULLTEXT01.pdf

[12] L. Söder och M. Amelin, "Effektiv drift och planering av kraftsystem," Stockholm, pp. 45–55, 2011. [Online]. Tillgänglig på: http://kth.diva-portal.org/smash/get/diva2:467466/FULLTEXT01.pdf

[13] L. Kuhlins. (2020, Mars) vattenkraft.info - info om svensk vattenkraft. Data för miljöorganisationer och diverse myndighetetr. [Online]. Tillgänglig på: https://vattenkraft.info/?alvid=286

[14] Energiföretagen, "Kraftläget i sverige - vattensituationen," pp. 1–2, April 2019. [Online]. Tillgänglig på: https://www.energiforetagen.se/globalassets/energiforetagen/statistik/kraftlaget/tidigare-kraftlagen/2019/kraftlaget-sverige-veckorapport-vecka-2019-16.pdf

[15] ——, "Kraftläget i sverige - vattensituationen," pp. 1–2, Okt. 2019. [Online]. Tillgänglig på: https://www.energiforetagen.se/globalassets/energiforetagen/statistik/kraftlaget/tidigare-kraftlagen/2019/kraftlaget-sverige-veckorapport-vecka-2019-41.pdf

CONTEXT E
HVDC GRIDS

POPULAR DESCRIPTION

The highway for current could help us prevent a climate doomsday

No one can ignore the fact that we are approaching a dead-end when it comes to mankind's handprint on the climate. Most of us are familiar with new and clean energy sources, but are less aware of the difficulty to connect them to the power cables coming to our homes.

The problem with incorporating more clean energy in the power grids is the distance to suitable locations where it is either windy, or sunny almost every day.

The distance problem has been recently solved by using the HVDC transmission line grid. The abbreviation stands for High Voltage Direct Current and could be described as the same type of voltage and current available in an ordinary battery, but at a much higher level. This is totally unlike the Alternating Current (AC) in the usual wall sockets in the homes.

As innovations had led to more effective and cheap equipment, HVDC has more and more begun to outcompete the old AC standard. This is already happening with some usages. For example, links are being built to massive offshore wind turbine farms at remote locations, where it was never profitable before. This trend is pointing towards a more strengthened HVDC power grid between continents in the future. Only then, the full benefits of wind and solar energy will be achieved. The issue regarding storage of overproduced energy in areas where all energy is not consumed will virtually forever be solved. This is because there will always be countries or areas where the wind is blowing, or the sun shines on. The HVDC “highway” grids will easily and with a minimum of losses bring the overproduced energy to areas with a shortage.

There is already an upswing of developments ongoing in renewable energy and its uses, but it will even be higher when big HVDC grids are built. The need then for carbon dioxide generating energy sources will drop significantly and could help save us from the doomsday of the climate.

SUMMARY OF PROJECT RESULTS

HVDC (High Voltage Direct Current) is an electrical power transmission system known as an electrical superhighway. Its functionality is to transmit a large amount of electrical power over long distances, and is nowadays an advanced tool for linking massive wind parks since it can deliver constant voltage level and frequency. The need for green energy in Europe has culminated in a projected potential of more than 100 Giga Watt in the North Sea.

In project E1 the main focus of the group is to analyze the occurring problems on controlling the power electronics in the converters, used to transform High Voltage Direct Current (HVDC) to High Voltage Alternate Current (HVAC). The conversion is important due to its ability to connect with an ordinary three-phase HVAC power grid. To be able to connect it is necessary to comply with the regulations of power quality. Those rules intend to prevent unnecessary disturbances in current and voltages contaminating the grid. Power losses and disturbances in power grids are always unwanted. The control signal for the converter is a way to affect those two variables. Different types of modulation strategies for the control signal are tested with help of simulation software to find out its behavior and the best performance.

In the following project E2, the project group aims to specify and develop fault detection algorithms through simulation for an HVDC network. For example, a ship is dropping its anchor on a submarine cable. The fault is detected by an algorithm that examines the voltage collapse and the current rise. The simulation for detecting the fault will be done in PSCAD, where two algorithms will be chosen for a good and accurate result.

As a result of project E1, higher frequencies of the wave are used to shape the control signal. This is done to minimize disturbances but by doing so there will also be a power loss in the converter switches. It becomes a trade-off.

A three-phase HVDC converter model was built in the simulation software Matlab Simulink, in which switches could be controlled by a signal modulated in several ways. The disturbances and the power losses in the converters were taken as a result. After several attempts, the result of the Total Harmonic Distortion (THD) showing a measure of the occurred disturbances. It became clear that the THD for the voltage was not improved by increasing the carrier frequency to shape the control signal. However, the THD for the current is improved at higher carrier frequencies. To be able to comply with the power quality regulation, harmonic filters have to be added. Unfortunately, project E1 could not be finished during the spring term and the report has been withdrawn.

In future studies for E1, filters can be used to filter out disturbances of alternating current if the control signal is not enough to be minimized. Adding filters would also increase the power losses. The elimination of harmonics interfering with the quality regulations could either be done by filtering out some of the harmonics after the usage of the converter, or by applying selective harmonics at the control signal to the converter switches.

Another way to get a better quality is to use a unipolar pulse width modulation. If two of those quality enhancing control signal techniques are combined, it would give a better result.

Project group E2 has introduced and evaluated the success of various simulated algorithms suitable for fault detection in the HVDC power grid using DC breakers. Another factor that was taken into consideration was the location and type of the fault in voltage collapse and current rise difference. This was done by the different algorithms.To be able to determine the most helpful algorithms to examine faults, a study on different algorithm characteristics had to be considered. Through applying the voltage derivative (dv/dt) and the traveling wave algorithm, the fault could be detected successfully. To successfully clear faults, a suitable threshold value has been selected, and when exceeded, a signal is sent to the HVDC-breaker to clear the fault.

In project E2 the focus is on protection and fault detection in the HVDC network. This contributes to safe and reliable protection for the converters by minimizing the overvoltage and overcurrent. Thereby, the lifespan of the equipment is greatly increased through less frequency of possible faults that usually cause damages. Combining the results of the two projects E1 and E2 will lead to further development and understanding of the HVDC network.

In future studies within the topic of project E2, a study can be done involving other algorithms for single-ended fault detection. For a better literature understanding, it is recommended to read through fault detection in double-ended algorithm detection such as *Directional Comparison* and *Longitudinal DC Line Current Differential.*

The reliability, high efficiency, and the system owner's regulations regarding the power quality are key elements when it comes to building a reliable HVDC system, otherwise, it would be neither economically nor practically justifiable. Both projects are two small steps in the right direction in a far more complicated technique to build a fully functioning, fault-free and optimized system.

IMPACT ON SOCIETY AND ENVIRONMENT

The demand on electricity is rising to incorporate more effective power transmission lines all around the world. It mostly has to do with the transition to cleaner and renewable energy sources to meet the climate goals. Indirectly, HVDC will have a big impact to make this possible by facilitating the connection of wind and solar farms at remote locations to the power grid.

A more suitable supply of energy to remote locations could further push economic development forward and can as well bring stability to some countries' power grids, where this previously has been a big problem.

However, an expanding HVDC industry will mean an increased production of manufacturing material for the technique. This immediately raises the question about in what circumstances the material is produced and how the work conditions are for the people involved in the process. The power electronics used in Europe are mostly produced in Switzerland under good working conditions, but when it comes to the material it is far more complicated to analyze.

Another problem with HVDC grids is that a political agreement has to be reached by the different countries on how the network can be used before applying. This can result in a better political climate between the different countries involved, as they are in need of cooperation. There is also a risk that despite an existing agreement, a country may decide to act as an insurgent and even to shut down a connection with a neighboring country. This could cause devastating power blackouts if the country itself does not have enough of its own energy production.

When it comes to the cost aspects, the DC lines used in the HVDC are cheaper than HVAC (AC) lines for long distances. The HVDC grids are the future because with each year more and more wind power stations will be built offshore.
One of the disadvantages that still remain with the HVDC grid is the grounding of the grid, as it is a very complex process and difficult to install. A counterargument is that HVDC grids are very useful as they can transmit power over long distances at lower costs. HVDC is also a very complicated and fragile transmission grid where a single fault or mistake in the converter or other components can lead to a change in the whole network which is expensive.

Neither installations nor operations are normally permitted in electrical power grid zones, simply to not jeopardize electrical safety. Power transmission in the HVDC for instance can have an indirect influence mostly on-site and course of land used for development.

When installing a new HVDC power tower grid in an area with a forest, the zone has to be without any obstacles, and especially trees. Also, the tower area has to be secured so no accidents can occur. This makes a lot of animals lose their shelters, yet, because of the line clearings for the grid, a habitat is often prepared for those species that could suffer because of the reduction of meadows.

The HVDC grid is an environment-friendly method used to avoid multiple cables, with no carbon dioxide emission. The usage of HVDC lines causes no polluted air that is safe for humans to inhale, as the sources of electricity on those grids are renewable.

To conclude, future advancements in the HVDC grid network face various threats but also numerous conveniences. HVDC, like most systems, has positive and negative aspects based on usage and application.

Yet, it is unavoidable to admit that the positive aspects of using HVDC and HVDC lines outweigh the negative aspects, especially when it comes to the social aspects, and specifically the positive impact HVDC technology has on the environment.

We see that HVDC grid networks will be used more in the future around the world, and capable of transmitting larger amounts of energy with less energy loss, i.e. much stronger and more developed.

Protection of HVDC Grids Against Blackouts (Simulation)

Amal Al-Ammari and Dinah Atchan

***Abstract*—High-Voltage Direct-Current (HVDC) grids are a promising technology used both offshore and onshore providing long-distance power transmission between different Alternating Current (AC) systems. The HVDC grid that is used offshore needs to be protected in many ways. The aim of this study is to define and create a fault detection system for an HVDC grid using a simulation software. Faults occur such, when for example a ship is lowering the anchor on a submarine cable. The algorithm will detect the fault by examining the voltage collapse and current increase. Yet another important factor that has to be taken into consideration is the location and property of the fault. The simulation is implemented in PSCAD software with two algorithms selected for a better and effective outcome. The results from both algorithms show that the detection of faults is faster the closer the faults are to the DC breaker.**

***Sammanfattning*—Högspänd likström (HVDC) är en lovande teknik som används både till havs och på land, det ger kraftöverföring mellan olika växelströmssystem. HVDC-nätet som används till havs behöver skyddas på många sätt. Syftet med denna studie är att definiera och skapa ett feldetekteringssystem för ett HVDC-nät med en simuleringsprogramvara. Ett fel som kan uppstå på elnätet är när till exempel ett fartyg råkar sänka ankaret på en sjökabel. Algoritmen kommer att upptäcka felet genom att undersöka spänningskollaps och strömökning. Ytterligare en viktig faktor som måste tas i beaktande är placeringen och egenskapen för felet i spänningsökningsskillnaden. Simuleringen implementeras i PSCAD-programvara med två algoritmer valda för ett bättre och effektivt resultat. Resultaten från båda algoritmerna visar att detekteringen av fel är snabbare ju närmare felen är till DC brytaren.**

***Index Terms*—High Voltage Direct Current (HVDC), Voltage derivative (dv/dt) algorithm, Traveling wave algorithm, PSCAD simulation, Multi-terminal grid, Fault detection, Threshold value, Offshore HVDC, DC breaker, Voltage collapse.**

Supervisor: Ilka Jahn

TRITA number: TRITA-EECS-EX-2021:157

I. Introduction

The rising demand for renewable energy has put new demands on the flexibility of transmission grids. These criteria, along with technological advancements in High-Voltage Direct-Current (HVDC), have culminated in the desire to develop large multi-terminal HVDC grids [1].

The HVDC technology has been established as a valuable enhancement to the existing alternative current (AC) system. The inherent properties of direct current (DC) make it possible for long cables to function with low losses. However, even with the lower losses, DC is still not widely used since transforming direct voltage was an almost impossible task due to the power plants that were first installed in the 19th century [2]. Due to the lack of equipment, building an HVDC was not possible but with today's modern converters it is now possible. HVDC is a more common technology now for attaching extensive wind farms where HVDC can offer on the AC side a constant voltage, frequency, and phase angle [1].

In Europe, the need for green energy has culminated in a projected potential of over 100 Gigawatts in the North Sea [3]. The generation from wind farms depends on the wind condition. The existing point-to-point HVDC connections are only fully utilized for short periods. If a wind farm were to be connected to several alternating current systems with few HVDC links, it would allow the electricity to keep flowing when the system does not operate at full power [1].

Characteristics that are important for the HVDC grid are grid regulation, converters, and protection issues. In the electricity grids, faults can always occur and therefore a protection system is needed to protect the grid and minimizing the disturbance. Without a protection system, a blackout can take place in the European HVDC grid [1].

The aim of this paper is to study and analyze different fault detection algorithms, giving the best result for fault detection in an HVDC grid. Through using the two selected algorithms, protection will be applied to the HVDC grid. This paper consists of the following: section II provides an overview of the fault and protection, section III describes the HVDC grid which is simulated in PSCAD, including the algorithms that are used for fault detection. In section IV the result of different simulations is presented and has been discussed in section V. The conclusion of this paper is found in section VI.

II. Background

When creating and using an HVDC grid, one of the most important factors is to have protection. In case of a blackout or fault in the grid, a detection has to be performed. The fault that is detected has to be cleared within a few milliseconds to prevent the grid from blackout [4]. In HVDC grids, protection is based on two components. The first one is fault detection and the second is fault clearing.

A. Protection requirements and philosophies

In the DC grid, there are six protection requirements:

- Sensitivity - an accurate detection of any fault that occurs in the grid.
- Selectivity - the protection system starts functioning only when a fault has been detected in its territory.
- Speed - the current of the fault must be cleared out before any damage is done to the electrical equipment.
- Reliability - in case of failure in the primary system, a backup system is used.

- Robustness - detection of faults in normal mode and degraded mode.
- Seamlessness - after clearing the fault, the system must continue operating.

There are three types of DC protection philosophies. These philosophies are used to protect the system from fault:

- Selective protection - part of the grid can be disconnected in case of fault detection.
- Partially selective protection - a zone of the DC grid is disconnected or at least one element that has a fault in the zone.
- Non-selective protection - the entire DC grid is disconnected when a fault is detected [5].

B. Fault types

Some faults can occur in the HVDC grid, such as when an anchor is set down, landing on the cable creating a fault. Two of the most common types of faults that are responsible for a faulty cable of the HVDC grid are line to ground and line to line [6].

Faults in the HVDC grid can cause a high current increase and voltage collapse. Therefore, the converters in the grid have to be protected against high currents in order to not damage the equipment.

C. HVDC fault detection

The DC fault is detected through an algorithm that examines the voltage collapse and the current rise. Depending on the location and property of the fault, the voltage collapse and current rise will be differing in amplitude.

There are two different types of methods that detect the fault in the grid, single-ended and double-ended. The single-ended method simply uses measurements from one line end. Double-ended methods use measurements from both ends of the line and need a communication channel [7]. In this paper, two algorithms were selected from the single-ended methods. The algorithms are voltage derivative (dv/dt) and traveling wave.

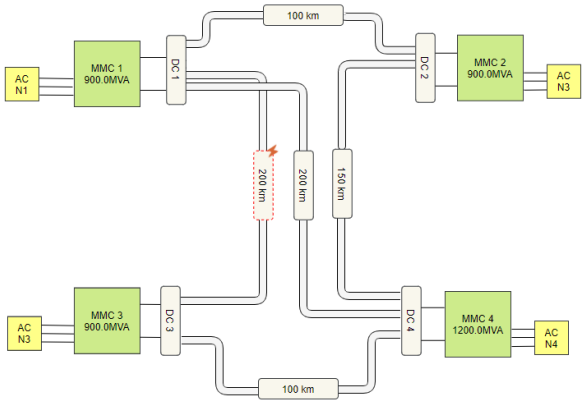


Fig. 1. HVDC test grid that is used in PSCAD. The fault is located in the red marked cable, link 13, with the length of 200 km.

III. METHOD

A. HVDC test grid

In Fig. 1, the HVDC test grid consists of four Modular Multilevel Converters (MMC) with different power sources connected through terminals. Converters 1 and 4 are connected to three links, but converters 2 and 3 are only connected to 2 links. The links are connected symmetrical and have the voltage of 320 kV. Each cable connected to the DC breaker has an inductance of 100 mH. The test grid is taken from [8].

The cables that are attached to the converter have different lengths. Cable 12 and 34 are 100 km long, while cable 13 and 14 are 200 km and cable 24 is 150 km long.

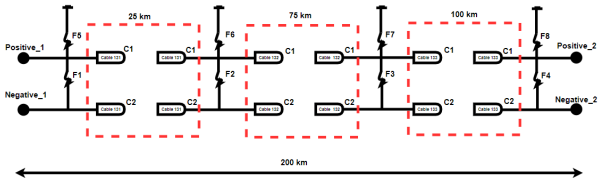


Fig. 2. An illustration of where the fault has occurred in different lengths of the cable. The positive and negative cables are parallel to each other. Faults (F1-F8) are the faults that are being investigated.

The investigated faults are located in cable 13 with different fault types simulated in PSCAD. When simulating the cable in link 13, it was divided into multiple pieces from the DC breaker which is shown in Fig. 2. This is described further in Table I and II. Depending on the location and the fault type that is detected in the simulated HVDC grid, pole to pole or pole to ground will be a short circuit.

TABLE I
CABLE LENGTHS WITH POLE TO POLE FAULT

Fault number	Fault location (km)
Fault 1	0
Fault 2	25
Fault 3	100
Fault 4	200

TABLE II
CABLE LENGTHS WITH POLE TO GROUND FAULT

Fault number	Fault location (km)
Fault 5	0
Fault 6	25
Fault 7	100
Fault 8	200

B. DC breaker

The DC breaker used in the following HVDC grid disconnects cables with fault. This is done by inserting a counter-voltage that reduces the fault current to zero. The DC breaker has a delay of 2 ms for opening when fault has been detected by fault detection algorithms. [9]. This time is equivalent to the opening time of a regular hybrid HVDC breaker with a mechanical disconnection to the HVDC grid.

C. Fault algorithm detection

To be able to further analyze the fault in the cable, two algorithms are selected and applied to the HVDC grid simulation. The two fault detection algorithms that were selected in this case were the voltage derivative (dv/dt) and traveling wave algorithms. Detection of the fault that is simulated in PSCAD was calculated in the positive pole on cable 13.

1) Voltage derivative algorithm: The voltage derivative algorithm, also known as dv/dt is the most discussed algorithm related to fault detection in DC grids. In case of fault, an initiated traveling wave by the fault causes the DC voltage to decrease and the DC current increase at 0.70 seconds.

This algorithm is also based on the traveling wave that is captured in the DC line protection. The DC voltage is sampled, and the derivative (dv/dt) is thereafter calculated.

One of the positive characteristics this algorithm has is the quick fault detection which is usually between 2-3 milliseconds [10]. The voltage derivative dv/dt stands for the difference between voltage (V) and time (t). The negative and positive poles are symmetrical which leads to equal trip signals.

The voltage is measured at two different time points ($t_2 - t_1$) which leads to a difference of 20 microseconds [11], calculated in equation 1. The threshold (V_{thr}) is set to 100 kV/s.

Equation 2 describes the trip criterion for the DC breaker. When the dv/dt signal is above the threshold, a fault has been detected, a trip signal is sent further to the breaker. Before the opening of the DC breaker, two milliseconds have to pass after the dv/dt has been triggered [9].

$$\frac{\delta V}{\delta t} = \frac{V(t_2) - V(t_1)}{t_2 - t_1} \tag{1}$$

$$|\frac{\delta V}{\delta t}| > V_{thr} \tag{2}$$

2) Traveling wave algorithm: Depending on where the fault is located in the cable, a traveling wave will be generated and sent in the opposite direction. An example of this is shown in Fig. 3 where the fault is located in (B).

The generated wave from the fault (B) will travel all the way to (A) and back. When the traveling wave has reached distance (A) and (B), a spiked pulse will be generated, and with each time the traveling wave goes back and forth the amplitude of the spike is reduced [11].

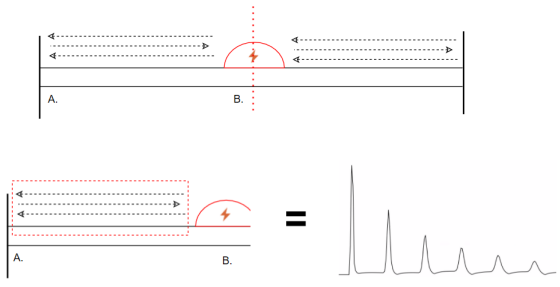


Fig. 3. An illustration of the traveling wave in a cable with a fault.

To detect and determine the value of the traveling wave that has occurred from the fault, the dv/dt algorithm and di/dt algorithm had to be combined and used. This is calculated as in equation 3. The value of the characteristic impedance of the wave is R. The same threshold value is used in the voltage derivative and traveling wave algorithm [12].

$$TW = \frac{\delta V}{\delta t} - R\frac{\delta I}{\delta t} \tag{3}$$

D. Extraction of data in PSCAD

The algorithms were simulated in the PSCAD software. After creating a schematic there was a possibility to save the data files with the variables from the simulation that was done. The output matrices were exported from PSCAD which were be used in MATLAB to draw graphs examining the voltage, voltage derivative, and traveling wave figures.

1) Simulation of dv/dt : The simulation of voltage derivative algorithm was simulated using different logical components. The delay component was used to calculate two different time intervals for the voltage. The round circle is a difference junction used to combine input signals. This gives a result which is then passed into the absolute value component. This is done so all voltage derivative values become positive as output in Fig. 4. The hysteresis buffer (triangle) is used to set a threshold value of 100 kV/s.

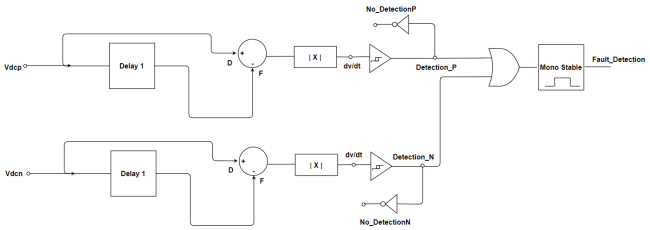


Fig. 4. A schematic of the dv/dt fault detection algorithm simulated in PSCAD using different logical components.

2) Simulation of traveling wave algorithm: The simulation of the traveling wave was carried out in the same way as shown in Fig. 4. This simulation has been modified to apply equation 3 for traveling waves. The impedance value of R as seen in Fig. 5 is 25 Ω. This PSCAD scheme is designed to calculate the traveling wave algorithm.

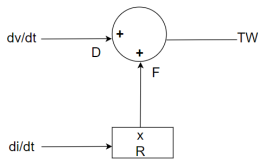


Fig. 5. Traveling wave schematic with a two-sided summing junction and one impedance (R) multiplier.

IV. Simulation Results

In this section, different graphs are presented with different results depending on the fault location and fault type. The fault is in cable 13 therefore a comparison between other none-faulted cables is done for cables 12 and 14.

A. Fault detection in cable 13

In Fig. 6 and Fig. 7, the voltage collapse is shown based on different lengths. The distance is calculated from the location of the DC breaker. The occurrence of the fault begins at 0.70 seconds, which can be seen in both figures. The voltage at this specific time begins to collapse where the DC voltage drops sharply from 320 kV to 0 V because of a short circuit between the negative and positive poles.

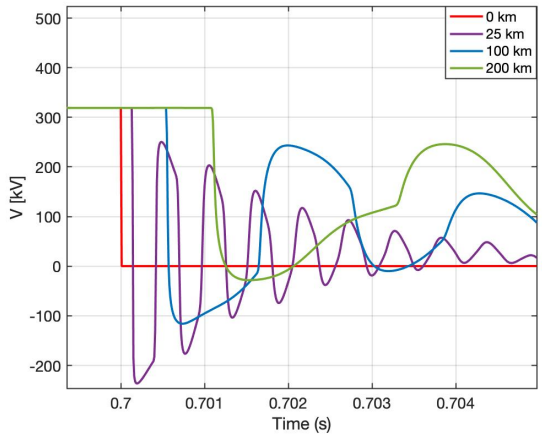


Fig. 6. Voltage for Pole to Ground (P-G) faults.

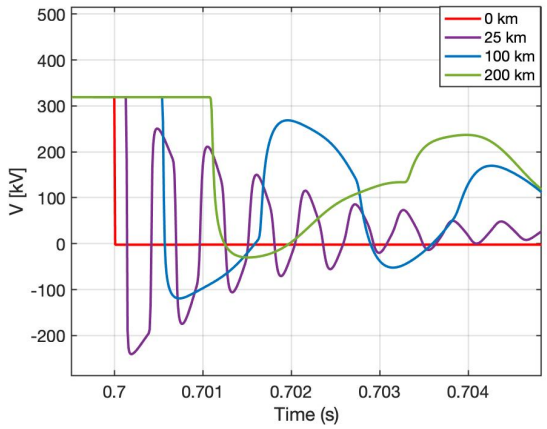


Fig. 7. Voltage for Pole to Pole (P-P) faults.

The algorithm output in the negative pole has the same characteristics as the positive pole but opposite. Pole to pole and pole to ground have almost the same characteristics when it comes to voltage collapse. However, the voltage collapse varies depending on the distance to the DC breaker. The closer the fault is to the DC breakers, the greater the voltage collapse. The fault that occurs at 25 km, has the largest voltage collapse, where it drops from 320 kV to minus 240 kV. The fault that is furthest from the DC breaker (200 km) has a lower voltage collapse compared to the faults that are closest to the DC breaker.

B. Comparison between faulted and none-faulted cables

In Fig. 8 the voltage in different cables is compared. The fault in cable 13 is located at 25 km away from the breaker while the other cables are not faulted. What all cables have in common is that they are connected to the same converter MMC1.

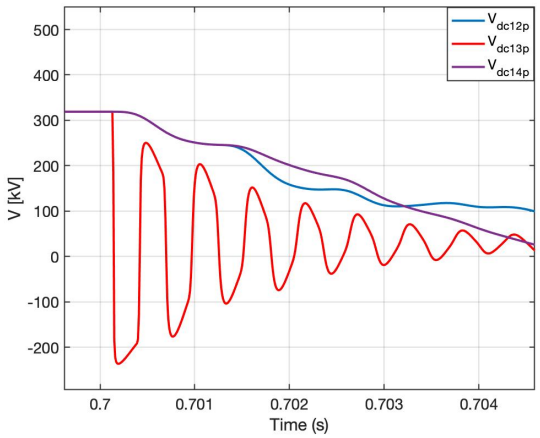


Fig. 8. A simulation comparison between cables 12, 13 and 14.

C. With and without opening of DC breaker

The fault detection is triggered by the voltage collapse which opens the DC breaker two milliseconds after the trip signal has been sent from the fault detection at 0.70 seconds. As seen in Fig. 9, the current in cable 13 is drastically increased from 0 to 35 kA. Fig. 10 is a graph of the current after the opening of the DC breaker and after the fault has been detected. In Fig. 10 When the DC breaker has opened, the current decreases and goes back to zero. In that case, the faulty cable 13 can be disconnected. Through using the algorithms for fault detection that are developed, the DC breaker will open normally after 2 ms.

D. Simulation of voltage derivative algorithm

The absolute value of the voltage derivative has been simulated in PSCAD to get a result of positive peak values, and the graph shows the maximum algorithm value. In Fig. 11, the voltage derivative in equation 2 is applied. Each spike in Fig. 11 is the derivative of the voltage with a time interval of 20 microseconds.

The highest peak derivative value is at fault 2, which is located 25 km away from the DC breaker. The further away from the fault and closer to the DC breaker at the converter terminal, the lower peaks in the voltage derivative graph. The threshold for the dv/dt algorithm is set to 100 kV/s. The high peaks are closer to converter 1.

E. dv/dt comparison between faulted and non-faulted cables

In the same way as earlier, the fault detection algorithm dv/dt is compared with the faulty and the non-faulty cables. Fig. 12 shows results for a fault on cable 13 with a maximum

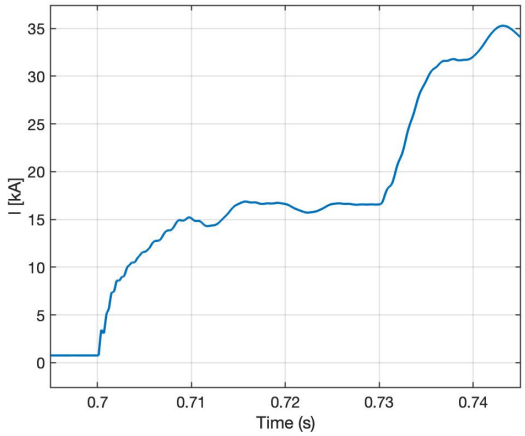


Fig. 9. Current without DC breaker opening.

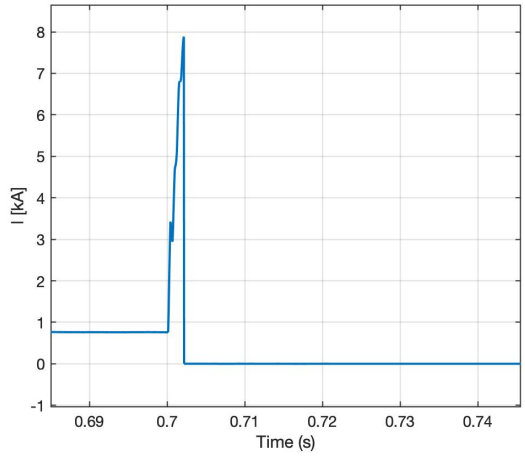


Fig. 10. Current with DC breaker opening.

derivative of 504 kV per 20 microseconds. Link 12 and link 14 do not have a fault which makes the derivative lower and not visible on the graph.

F. Simulation of traveling wave algorithm in link 13

When simulating the traveling wave algorithm, the distances (0, 25, 100, and 200 km) have a low di/dt value. The oscillations are the same as for the voltage derivative, but the peak values are higher in the traveling wave in Fig. 13. The traveling wave algorithm using both the current and the voltage derivative is calculated using equation 3. The threshold value is 100 kV/s as seen in Fig. 13. The fault detection in this graph is of the type of pole to ground (P-G). When the fault occurs at 0.70 seconds, 10 μs later the traveling wave algorithm will detect the fault. The maximum peak at 25 km has a maximum value of 506 kV/20 μs. When the cable is located further away from the DC breaker and closer to the converter 1, the peaks will have a lower amplitude.

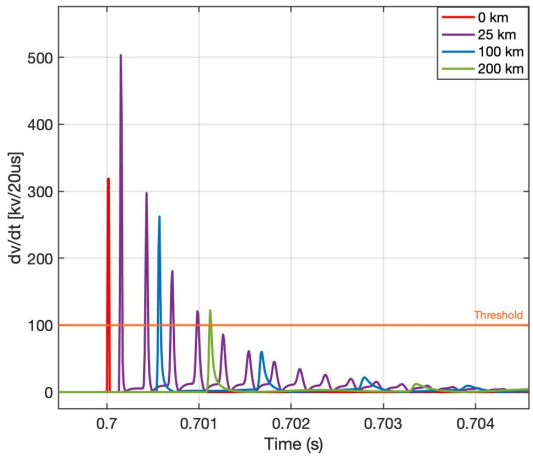


Fig. 11. Voltage derivative at different fault distances.

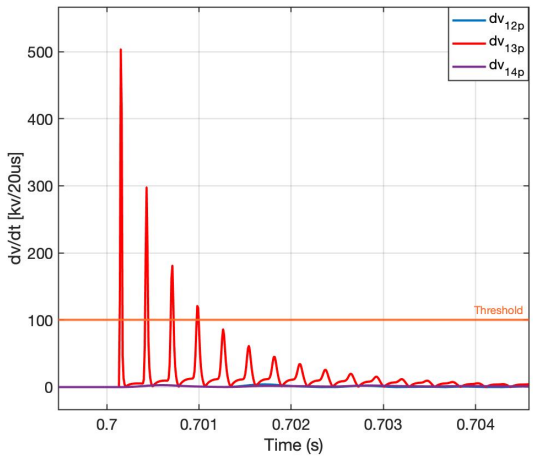


Fig. 12. Comparison between the voltage derivative of the faulted and non-faulted cables.

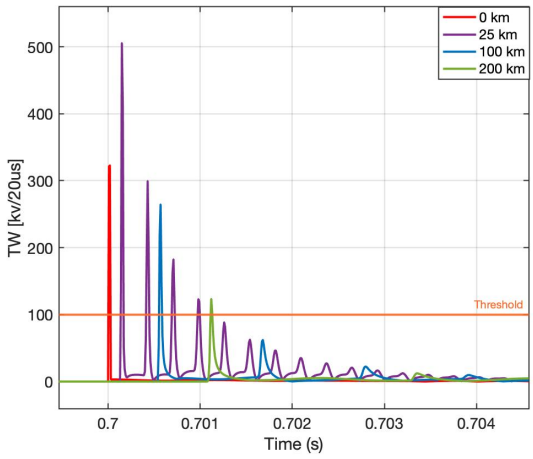


Fig. 13. Traveling wave with different fault distances.

V. Discussion

A. Analysis of DC fault detection

When a fault occurs, there will be a voltage collapse in the cable, in this case cable 13. The result of the cable voltage

will vary depending on the distance from the DC breaker. In Fig. 6 and Fig. 7 a comparison is made with pole to pole and pole to ground faults for different fault distances. In the pole to ground case, all faults occur at 0.70 seconds and it has a symmetrical voltage drop with the type pole to pole. Faults 1 and 5 that are located in the 0 km from DC breaker and have a drop to 0 V. This is because of the faults are located at the beginning of the cable. The fault at 25 km, compared with 100 and 200 km, has the highest voltage collapse because it is the closest to the DC breaker. The voltage collapse at 200 km is lower, with a wider oscillation. This is due to the fault that has occurred at the end of the cable. When the fault is at 25 km, the time of the voltage drop is 0.14 ms. The fault at 100 km has a time of 0.54 ms while when the distance is 200 km, the time is 1.11 ms. The duration of the collapse for each fault is different depending on the distance of the fault in the cable.

In Fig. 6 and Fig. 7 the waveform becomes wider, the longer the distance is from the DC breaker, which makes the voltage collapse smaller. This is due to the energy loss when the wave has traveled across the cable and the impedance value of the cable.

In Fig. 8, a comparison is done between cable 13 with a fault and cables 12 and 14 without a fault. In cables 12, 13, and 14, a voltage collapse takes place but goes much faster in cable 13. In cable 13 there is a fault where the voltage drops from stable, sharply to negative. The fault occurs 25 km away from the DC breaker. The inductance that is connected with the DC bus 1 effects both the voltage and the faulty cable by damping the oscillations. The damping through the inductance is high, therefore the signal is lower/weaker in cables 12 and 14. Cable 13 has a quicker dampening on oscillations with higher amplitude. The difference between cable 13 compared with cables 12 and 14 is the high inductance that removes fast oscillations.

B. Analysis of dv/dt fault detection algorithm

The threshold value in the dv/dt algorithm as shown in Fig. 11 has the value of 100 kV/s. The lowest value of the voltage derivative 200 km away from DC breaker is 123 kV/s, which results in a suitable value for the threshold. The protection in the DC cable works when the fault detection derivatives exceed the threshold value. From Fig. 11, the derivative's maximum amplitude in different lengths based on 0 km, 25 km, 100 km, and 200 km exceeds the threshold value. The maximum value for dv/dt for the fault at 25 km has the value 504 kV/20 μs. This is the maximum value because it is closest to the DC breaker. As seen in Fig. 11, the longer the distance is, the lower amplitude for each peak. When the fault is located further away from the DC breaker, the maximum value of the dv/dt algorithm will be lowest in faulted cable 13, this is because the impedance is higher, which will dampen the fault wave. When the dv/dt algorithm exceeds the threshold value, the protection algorithm trips the DC breaker.

A comparison between the application of the dv/dt algorithm can be seen in Fig. 12, where cable 13 has errors at 25 km while cables 12 and 14 have no-fault. The peak of cable 12 and 14 of the derivative is not visible in the graph because of the small values. This is because the voltage derivative on the cables without fault is approximately 4 kV/20 μs, which is very low in comparison with the cable that has a fault with the highest voltage derivative of 504 kV/20 μs. This is due to the fault location and value of the impedance for the DC cable. The faulty cable has a faster damping and higher derivative peak. The large inductance value eliminates rapid peaks in cables 12 and 14.

C. Analysis of traveling wave algorithm

The same principle used in the dv/dt algorithm is applied in the traveling wave algorithm, where a current derivative di/dt, is taken into consideration. The difference between the two samplings is calculated and if the result is greater than the threshold, the fault will be detected.

In Fig. 13, the longer the distance, the lower peaks of the traveling wave are generated. The wave reflection of the traveling wave will become smaller in the faulty cable with time. This leads to the traveling wave decreasing in amplitude which will not be seen at the end of the cable. The traveling wave has a stronger signal than the dv/dt algorithm. The maximum peaks in dv/dt are lower than the traveling waves. This is because of the difference between dv/dt and di/dt in equation 3. Taking the dv/dt and di/dt into consideration would lead to a stronger fault detection signal for the HVDC grid.

VI. Conclusion

This paper presented a fault detection and protection strategy that protects the simulated HVDC grid. This is done by using two algorithms from the single-ended fault detection methods, voltage derivative, and traveling wave.

Using the dv/dt and traveling wave algorithm, the fault can be detected quickly with a delay of a few tens of microseconds. Through simulating the difference in voltage collapse for different fault locations can be measured. The voltage difference between the fault voltage is -207 kV for pole to ground in Fig. 7. The voltage difference between dv/dt is 381 kV/s. By utilizing fast fault detection using the dv/dt algorithm, the DC cables that are connected to the same converter can be protected.

As for the traveling wave algorithms the difference between the first maximum peak and the peak on the 200 km fault is 382 kV/s. Through using the traveling wave algorithm, the fault can be detected by taking the current derivative into account, which leads to a better result of protection for the HVDC grid.

One of the common things both algorithms have is the quick detection of fault accurately. The traveling wave has a higher peak value, which leads to better and more precise fault detection. Detecting the fault quickly and locating the fault using the dv/dt algorithm, various disturbances can be avoided, and components being destroyed due to the fault. If the detection is not done efficiently and quickly, the converter and cables will be destroyed. Cables with long distances lead to more difficult fault detection even if the algorithms are

applied. This is due to the wave dampening decreasing the wave coming from the fault.

To detect the fault and protect the HVDC grid, both sides of the cable must be considered. In future studies, double-ended fault detection can be taken into consideration, where the fault can be detected from two sides for better grid protection.

ACKNOWLEDGMENT

We would like to express our heartfelt gratitude to our supervisor Ilka Jahn for her time, guidance, determination, and dedication.

REFERENCES

[1] I. Jahn, "Context E HVDC grids, KEX projektvalskatalog VT21," KTH, Stockholm,Sweden, pp. 22–24, 2021.

[2] N. Stenberg, "The impact of hvdc innovations on the power industry," KTH, Stockholm,Sweden, 2013. [Online]. Available: http://kth.diva-portal.org/smash/record.jsf?pid=diva2%3A626845&dswid=-585

[3] J. Wassink. (2012, Apr) Power transport underestimated. Delta, Journalistic platform TU Delft. [Online]. Available: https://www.delta.tudelft.nl/article/power-transport-underestimated#.

[4] M. Mobarrez, S. Acharya, and S. Bhattacharya, "Impact of dc side fault protection on performance and operation of multi-terminal dc (mtdc) systems," in *2018 Thirteenth International Conference on Ecological Vehicles and Renewable Energies (EVER)*, 2018, pp. 1–7.

[5] J.-B. Curis, J. Descloux, N. Grisey, A. Wagner, Ö. Göksu, O. Saborío-Romano, C. Brantl, M. Kaiser, M. Quester, P. Ruffing *et al.*, "Deliverable 1.3: Synthesis of available studies on offshore meshed hvdc grids," 2016.

[6] V. Nougain, S. Mishra, G. S. Misyris, and S. Chatzivasileiadis, "Multi-terminal dc fault identification for mmc-hvdc systems based on modal analysis -a localized protection scheme," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, in press, 2021+.

[7] I. Jahn, N. Johannesson, and S. Norrga, "Survey of methods for selective dc fault detection in mtdc grids," in *13th IET International Conference on AC and DC Power Transmission (ACDC 2017)*, 2017, pp. 1–7.

[8] W. Leterme, N. Ahmed, J. Beerten, L. Ängquist, D. V. Hertem, and S. Norrga, "A new hvdc grid test system for hvdc grid dynamics and protection studies in emt-type software," in *11th IET International Conference on AC and DC Power Transmission*, 2015, pp. 1–7.

[9] M. Callavik, A. Blomberg, J. Häfner, and B. Jacobson, "The hybrid hvdc breaker," *ABB Grid Systems Technical Paper*, vol. 361, pp. 143–152, 2012.

[10] D. Naidoo and N. Ijumba, "Hvdc line protection for the proposed future hvdc systems," in *2004 International Conference on Power System Technology, 2004. PowerCon 2004.*, vol. 2, 2004, pp. 1327–1332 Vol.2.

[11] L. Yue, "Research on hvdc single-ended fault traveling wave location method based on fastica," in *2016 International Conference on Audio, Language and Image Processing (ICALIP)*, 2016, pp. 509–514.

[12] I. Jahn, F. Hohn, G. Chaffey, and S. Norrga, "An open-source protection ied for research and education in multiterminal hvdc grids," *IEEE Transactions on Power Systems*, vol. 35, no. 4, pp. 2949–2958, 2020.

CONTEXT F
POWER SYSTEM CONTROL

POPULAR DESCRIPTION

Saving our planet at the cost of our morning coffee?

Does sustainable energy create unsustainable electrical grids? The push for green energy might spell disaster for your morning routine, when solar cells crash the grid. Unlike coal power plants, renewable energy has no emissions, but is it worth breathing fresh air if you can't enjoy a warm mocha while checking your Instagram feed? Is there a way we can save the planet and enjoy our delicious coffee at the same time?

Traditional energy sources produce power by splitting atoms or burning fossil fuels. These sources also generate momentum by rotating generators, something that keeps the electrical grids stable. Without it, the grids cannot handle large spikes of usage, resulting in annoying lightbulb flickering and electrical machines running unevenly. Worse things happen when part of the grid breaks, it might trigger a cascading failure, leading to a complete black-out in a region. Renewable energy sources like solar panels and wind power lack this momentum, due to the sudden changes in weather conditions. When it gets cloudy or the wind stops blowing, the energy production stops almost immediately. By adding artificial momentum to renewable energy sources, we can avoid brown-outs and make sure you get the power you need for your daily needs.

Adding momentum is one way of saving the grid, another is adding specialized stabilizers, technology that takes unstable power, modifies it and sends out stable power. Momentum and stabilizers are key elements to allowing you to have your own smaller version of the grid at home, a microgrid. This allows you to generate your own electricity at home, making you less reliant on the larger grid. If you generate more electricity than you need, you can sell the surplus back to the electrical companies.

So don’t fret, future solutions ensure warm morning coffee whilst saving the planet. Renewable energy creates new challenges in the journey for a green future. Challenges that engineers are tirelessly solving.

SUMMARY OF PROJECT RESULTS

The power system is currently being developed to include large amounts of renewable energy sources that replaces more traditional large, generating units, previously forming the backbone of the system. These changes place new demands on the power system partly due to characteristics of renewable energy sources but also the location of said power sources.

This creates new demands on the electrical grid, both at transmission and distribution level, demands such as limits on the stability of the system due to less rotating mass in the system and increased variation in voltage, power flow and frequency. This requires new, efficient controls such as automation systems and controllable power system components to meet the new demands of the power systems.

The project group in F1 has investigated the effect of load disturbances in varying inertia systems. The model for investigation was set to replicate the tendencies in the Nordic electric grid and its shift towards being a lower inertia system. A lower inertia system means less rotating mass in the system, which is caused by the replacement of non-renewable energy resources as oil and nuclear plants for renewable energy sources such as wind and solar power. This replacement poses a challenge on frequency control, as the total system mass is lowered, contributing towards a more load disturbance sensitive system. The aim of the project was to understand the challenges following these tendencies and to provide a solution for frequency containment in a low inertia system, using batteries as a means to implement artificial inertia.

In future projects, further investigation on the implementation of battery systems for frequency containment focusing on the implementation is needed. A more thorough investigation on the challenges and costs of implementing and running these systems for frequency containment purposes are needed.

The project group F2 has tuned a Power System Stabilizer (PSS) in a system that is small signal unstable. When a system is exposed to small disturbances which cannot be suppressed, the oscillations increase causing the system to become small signal unstable. The small signal stability of the system provides insight into how a system can react to small disturbances from deviations in power, which can be dampened out or lead to instability. As systems shift towards a greater suppliance from intermittent renewable energy sources more power variations will occur. This will cause small system instability resulting in power outages. Through the use of power system stabilizers these power variations can be dampened out.

The next step for future studies would be to look at how a system would react to a change in its energy supply from a low amount of intermittent renewable energy sources to a large amount. The focus would then be on whether or not the resulting instabilities could be tuned with a PSS.

The group in project F3 has combined renewable energy sources with adaptive battery storage to create a sustainable and future-proof residential microgrid. Using Simulink, the group has created stable solutions for maintaining grid voltages under different load and environmental conditions. When required, the hybrid microgrid has the capability to accept input from the main grid, as well as outputting energy for monetary gain. The sudden drop in voltages and rapid change in loads can cause instability and undesirable isolations in the system. In order to avoid such problems, and guarantee a stable electrical system, a stable control system has been developed to deal with different generation and consuming conditions. Maximum power point tracking algorithm (MPPT) was also developed for a photovoltaic converter to maximize power generation.

In future projects concerning microgrids, an automated decision making algorithm that is capable of selling the stored power to the utility grid when surplus, and furthermore charge the battery when the spot price is cheap, which can be tracked using Nord Pool.

IMPACT ON SOCIETY AND ENVIRONMENT

Electricity affects the society on an environmental and economical level and is essential to modern society. A shift towards renewable energy sources will have an impact on society and environment. Global energy demand is expected to increase simultaneously as the world undergoes a shift towards renewable energy sources. This will cause problems on the stability of the electrical grid. Solutions like battery packs, residential microgrids, and power system stabilizers have the potential to provide electricity to individuals who lack access to a dependable energy source.

As countries all over the world shift their focus towards a greener future, the demand for renewable energy sources increases. As more underdeveloped countries move towards developed societies, the energy required globally is expected to increase. Therefore, stable and reliable electricity is vital in the shift towards a functioning modern society. This in turn creates a demand on the current and future generation of engineers to develop new technical solutions to reach these new goals. A solution to the stability problem is the introduction of artificial inertia to the grid. This may be done by large battery parks, which in turn have an effect on the environment, as the manufacturing process requires environmentally damaging materials.

The introduction of renewable energy sources require various adaptations of the current electrical grid to ensure a stable and reliable source of electricity. This implies great economic investments into the current grid. Also, future work on the power grid may put strain on the natural habitat. Doing work on transmission lines, far outside the cities interferes with the wildlife fauna. Renewable energy sources also damage the natural habitats of wildlife fauna. For example birds have been seen flying into wind farm rotor blades and getting crushed. This requires careful consideration of where to place these renewable energy sources, as not to endanger any species.

With new developments in micro grid technologies, people can empower themselves by investing in solutions that will supply power for their own home, or their small communities. Whilst the initial costs may be high, most microgrid solutions will be financially sustainable within a few years of the purchase. Not only will their home be self-sufficient, they might even be able

to sell electricity back to the main grid, offsetting their costs even more. The benefits are many, and as more installations occur, new financial options will develop. One such option might be that the banks can loan you money specifically for a micro grid, where you can pay off the interest with the savings you gain by having the grid installed. The microgrid system depends mainly on the battery which is the main part of the storage system, and the solar panel which represents the renewable energy source in the system. These parts are both highly recyclable, and while there are few recycling systems in place today, the market is looking at providing these services to individuals. Battery companies are currently investing in recycling stations that completes the life cycle of their batteries, as it provides a way to recycle almost all parts of their batteries. This provides a closed loop product lifecycle between company and individual, which helps to heighten efficiencies and reduce end-user costs.

To make sure the main grid can support many individual micro grids, some kind of regulatory control mechanism must be implemented, so that the main grid stability can be guaranteed. If the microgrids produce too much power, the grid owner must be allowed to shut off sell-back capabilities so the grid doesn't overload. This will impact microgrid owners negatively as they might be dependent on that revenue income. There is also the risk that these capabilities will be far-reaching and might impact the privacy of individuals, by letting companies control your electrical capabilities with even finer granularity. They harvest this information and sell it to other companies who can then profile you.

The problems for a greener future are many and intricate. There are many aspects to be taken into consideration to ensure the best possible outcome for future generations. Solving these challenges will take effort from engineers, companies and governments all working united towards a common goal.

Frequency Stability in Future Low Inertia Power Systems With Battery Support

Emil Bergvall and Alessandro Bonetti

***Abstract*—In the search for green energy to combat climate change, a shift from conventional energy sources such as coal, oil, and nuclear towards Renewable Energy Sources (RES) is needed. This shift poses a threat to the stability of the power grids as RES do not contribute with rotating mass in the system. A lack of rotating mass, or in other words inertia, jeopardizes the ability of power systems to counteract large disturbances. Frequency Containment Reserves (FCR) units are responsible for controlling the frequency in power systems by regulating the balance between the generated and consumed power. If the frequency deviates outside of the defined range from the nominal value, it can lead to system separation, blackouts, and system equipment damage. The frequency deviations are faster in low inertia systems, making it more difficult for FCR to keep the frequency within accepted ranges. Hydro turbines are often used as FCR units, but additional means of support could be needed for low inertia systems. Viable support could be battery systems. This project investigates the change towards low inertia and the possible implementation of a battery system as fast step-wise power support with a frequency trigger. The investigation is done through case studies of simulated system models in Matlab and Simulink.**

***Sammanfattning*—I jakten på grön energi för att bekämpa klimatförändringarna behövs en övergång från konventionella energikällor som kol, olja och kärnkraft mot förnyelsebara energikällor. Denna övergång utgör ett hot mot kraftnätens stabilitet då förnyelsebara energikällor inte bidrar med roterande massa. Brist på roterande massa eller med andra ord tröghet äventyrar kraftsystemens förmåga att motverka stora störningar. Frequency Containment Reserves (FCR) är system som aktivt arbetar med att styra frekvensen i kraftsystemet genom att reglera balansen mellan den producerade och konsumerade effekten. Om detta misslyckas och frekvensen avviker för mycket från den nominella frekvensen kan detta leda till systemseparation, strömavbrott eller skada hos systemkomponenter. I ett system med låg tröghet blir frekvensavvikelserna snabbare. Detta gör det svårare att använda sig av FCR för att hålla frekvensen inom accepterade intervall. Vattenkraftverk används ofta som FCR enheter, men för system med låg tröghet kan ytterliggare stöd behövas. Ett möjligt effektstöd kan vara batterisystem. Detta projekt undersöker förändringen till lägre tröghet i ett kraftsystem och möjlig implementering av ett batterisystem med ett snabbt stegsvar för effektstöd, vilket aktiveras vid en förbestämd frekvens. Undersökningen görs genom studier av specifika fall med en linjäriserad modell av ett kraftsystemet, simulerade i Matlab och Simulink.**

***Index Terms*—Battery power support, Frequency Containment Reserves, Fast Frequency Reserves, Low inertia, Renewable Energy Sources**

Supervisor: Danilo Obradović

TRITA number: TRITA-EECS-EX-2021:158

I. Introduction

The world is searching for green solutions and green energy is a step in that direction. The shift from conventional energy sources such as coal, oil, and nuclear towards Renewable Energy Sources (RES), like wind and solar, are currently under way in ravaging speeds [1]. Both wind and solar can be classified as RES as they follow the definition given by [2], being naturally replenishing and virtually impossible to exhaust, although they are limited in the amount of energy they can provide during a certain time, making them a preferred choice of energy.

Modern power systems have a set nominal frequency which, depending on the area, is usually either 50 or 60 Hz, with the Nordic system being a 50 Hz system. If this frequency deviates too much from its nominal value due to disturbances such as large load changes or a generator failure, the consequences could result in damage to the power grid. To protect the power systems if such an event happens, there are many triggering protection systems. Some examples are Under Frequency Load Shedding (UFLS), system separation and steam turbine shut down as stated in [3]. However, there are units actively working to keep the frequency within nominal ranges. Such units are called Frequency Containment Reserves (FCR). These units regulate their generated power to balance the frequency in case of disturbances. Traditional power systems have inertia in the system, providing stability through stored kinetic energy from the rotational mass. This extends the time period to react to a frequency deviation allowing FCR units the possibility to compensate for the disturbance.

Classical power generators produce energy through mechanical rotation, providing inertia to the system in the process. They are generally controllable, meaning that the operator of the power source can change the power output depending on the need. RES are controllable as well although usually only possible to reduce the power output. In contrast RES are generally decoupled from the power system through power converters as stated in [4], meaning that the power sources do not have a direct mechanical connection to the system meaning no inertia contribution. The lack of contributed rotating mass results in a net zero stored kinetic energy. Which, results in a zero inertia contribution toward the system. If classical power generators are replaced by RES, a loss of rotating mass occurs in the system, reducing frequency stability in the power system. For the current N-1 Nordic grid this change means that large disturbances to the power system result in greater deviations, possibly causing blackouts. N-1 is a criteria, stating that the system should be operable even if a loss of one large

generating unit occurs, thus running with N-1 generating unit in the system.

For a sustainable future, the integration of RES is needed. Hence the need for a solution, making the integration possible. This project studied the gradual replacement of traditional generators with RES and its impact on the frequency stability of the power system. The study was done by using Matlab to simulate a linearized power system implemented in Simulink. The power system had an additional Battery Energy Storage System (BESS) for fast power support. The power system was then gradually altered by replacing traditional generating units with RES, thus lowering the system inertia. Ultimately, the study resulted in a proposed integration of BESS as a viable solution for the integration of RES.

II. System Model

To investigate the impact of a load disturbance in an electrical power system, a linearized model with four cases of inertia level is set up in Simulink. This model simulates the effect of a disturbance in load or production balance, leading to frequency deviations. The four cases are set to replicate the tendencies of large electrical grids, where nuclear plants are gradually replaced with RES [4]. The four cases are shown in Table I. The model utilizes the hydro units as FCR units, meaning that their purpose is to regulate the frequency of the system. This situation mirrors the Nordic synchronous area, where hydro power units are commonly used as FCR [3]. Each generating unit is set to have different inertia time constants M. Hydro unit 1 is set to $M_1 = 8$ s, Hydro unit 2 is set to $M_2 = 6$ s, Nuclear unit 1 is set to $M_3 = 10$ s and Nuclear unit 2 is set to $M_4 = 12$ s. The first case is used as a base case, where both nuclear units are connected to the grid. Then, for each following case, one nuclear unit is replaced by a RES unit. Ultimately, for case 4, both nuclear units are replaced by RES units. For this case, a supplementary battery power support is introduced as a Fast Frequency Reserve (FFR). The purpose of FFR is to serves as a complement for the FCR units in low inertia situations, working as a first measure to mitigate a large disturbance [5]. The load disturbance ΔP_L is modeled as a step change, activating at a set time. For simplicity, the step is distributed to the two hydro buses equally.

TABLE I
The four system model cases, showing the amount of Hydro, Nuclear or RES units used. For each case, the total system inertia is displayed.

Case	Hydro	Nuclear	RES	System inertia M [s]
1	2	2	0	36
2	2	1	1	26
3	2	1	1	24
4	2	0	2	14

A single line representation of the first case model is shown in Fig. 1. The load disturbance is shared by the two hydro units at two places, as shown in Fig. 1. A single line representation of the last case model, case 4, is shown in Fig. 2. Like the first case, the load disturbance is shared equally by the two hydro units, as shown in Fig. 2, but the system also has access to a battery system for additional FFR support, in the diagram shown as ΔP_B.

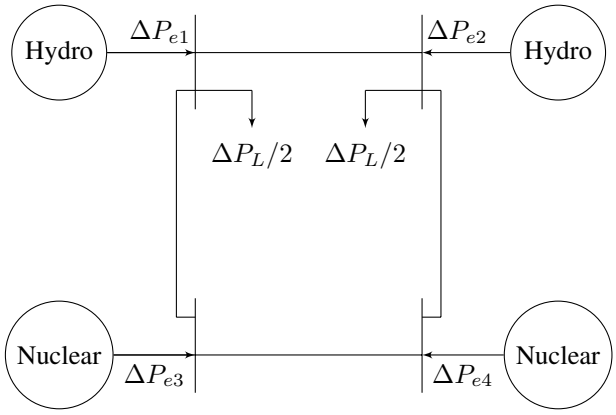


Fig. 1. Case 1, two hydro units and two nuclear units.

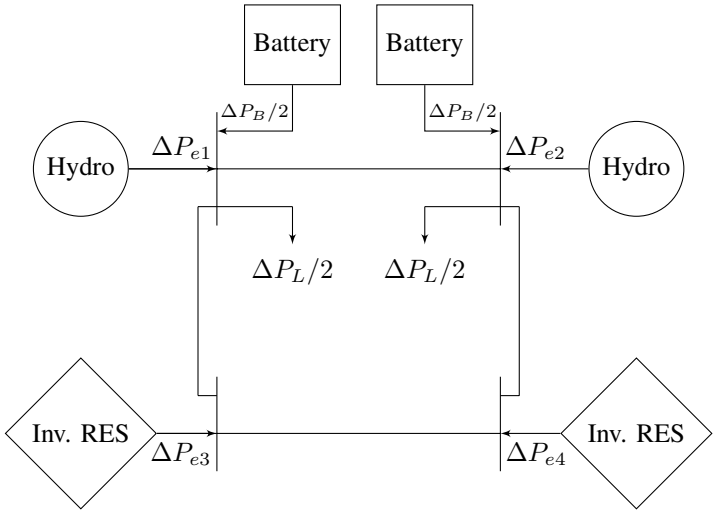


Fig. 2. Case 4, two hydro units and two inverter based renewable energy source (RES) units and the additional battery support.

The rotor dynamics of synchronous machines are given by the swing equation. Since the system is linear, deviations Δ from steady state are considered, the swing equation for an ith synchronous machine, using the dot notation for the time derivative is given by

$$M_i \Delta\dot{\omega}_i = \Delta P_{mi} - \Delta P_{ei} \tag{1}$$

where $\Delta\omega_i$ is the deviation in rotor speed from its nominal value in [pu], M_i is the inertia time constant in [s], ΔP_{mi} is the deviation of the mechanical power in [pu] and ΔP_{ei} is the deviation of the electrical power in [pu] for an ith synchronous machine.

Considering the Center of Inertia (COI), which is a reference frame considering all generating units in the power system as one, the swing equation is given by

$$M\Delta\dot{\omega}_{COI} = \Delta P_m - \Delta P_e \tag{2}$$

where M is the total inertia in the system, referred to as the system inertia. Furthermore the total mechanical power P_m

can be regarded as the total generated power, while the total electrical power P_e can be regarded as the total consumed power, the load. Assuming that the load is composite, the total electrical power P_e is modeled as both a frequency-dependent and non-frequency-dependent load by

$$\Delta P_e = \Delta P_L + D_{COI}\Delta\omega_{COI} \tag{3}$$

where ΔP_L is the non-frequency-dependent load change and for a positive damping constant D_{COI}, $D_{COI}\Delta\omega_{COI}$ is the frequency-dependent load change. Hence the swing equation for a load disturbance regarding the COI is given by

$$M\Delta\dot{\omega}_{COI} = \Delta P_m - \Delta P_L - D_{COI}\Delta\omega. \tag{4}$$

The disturbance will be counteracted by the two hydro FCR units. Since the load step change is distributed equally by the FCR units, the damping constant D_i for an ith hydro unit is set to be half of the COI damping. That is

$$D_i = \frac{1}{2}D_{COI}. \tag{5}$$

For simplicity, the FCR units in the system model are set to have the same parameter values for all parameters except the inertia time constant M. The model utilizes two FCR units, which have access to additional support ΔP_B. This support is divided between the two FCR units equally, thus the support for one unit is $\Delta P_B/2$. Given the single line representation of the case models, see Fig. 1 and Fig. 2, the FCR units are connected to the grid, resulting in a power flow ΔP_{ei} to the bus. Therefore, the swing equation for the ith FCR unit is given by

$$M_i\Delta\dot{\omega}_i = \Delta P_{mi} + \Delta P_B/2 - \Delta P_{ei} - \Delta P_{Li} - D\Delta\omega_i \tag{6}$$

hence, the Rate of Change of Frequency (RoCoF) is proportional to the power unbalance. The maximum RoCoF is however, also inversely proportional to the inertia.

In the model, the hydro units are utilized for FCR purposes, hence the change of mechanical power ΔP_m in the system is only given by the Hydro units. The hydro unit utilized in the system model is shown as a block diagram in Fig. 3. The figure shows the turbine governor, which serves the purpose of regulating the control gate of the turbine such that power is produced as desired. The dynamics used to model the hydro unit are derived by Bernoulli's equation in [3]. Considering the transfer block shown in the figure, the hydro dynamics are non-minimum phase. That is, a zero is in the right half plane. This property results in an initial decrease of produced power following a step change of increased power. This is due to the lower pressure caused in the tubes following a sudden change of the control gate [3]. Non-minimum phase systems are harder to control, due to the behavior previously presented results in an undershoot directly following a step change. This may be considered as an effect similar to delay in the response (to some approximation). Hence, to provide additional fast and immediate support, a battery system is added to the hydro bus.

The nuclear unit utilized in the system model is shown as block diagram in Fig. 4. As the nuclear units will not be used for FCR purposes, the mechanical (generated) power will remain at steady state. Thus their dynamics are fully determined by the swing equation given in equation (1) with $\Delta P_{mi} = 0$.

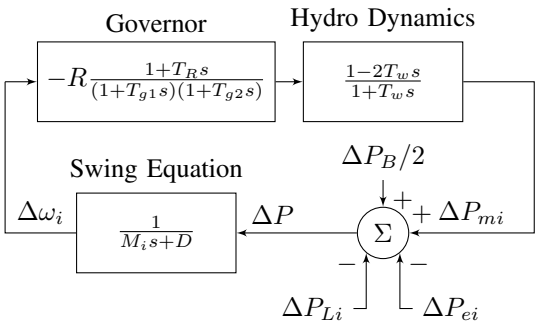


Fig. 3. Block diagram of an ith hydro unit used as FCR with battery support available, visualizing the governor block, hydro dynamic block and the swing equation block.

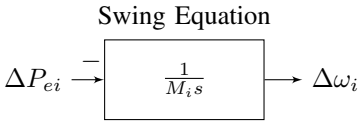


Fig. 4. Block diagram of an ith nuclear unit, visualizing the swing equation block without damping. The nuclear units are not used for frequency regulations, hence, their mechanical power will remain at steady state. Thus the deviation $\Delta P_{mi} = 0$.

All parameters shown in the block diagrams in Fig. 3 and Fig. 4 are described in Table II.

TABLE II
SYSTEM MODEL PARAMETERS

Parameter	Description	Unit
M	Total system inertia time constant	s
M_i	Individual unit inertia time constant	s
D_{COI}	System damping constant	pu
D	FCR unit damping constant	pu
T_w	Hydro dynamic time constant	s
R	Governor system gain	pu
T_R	Transient feedback loop time constant	s
T_G	Main hydro servo time constant	s
g_{st}	Speed-droop	pu
g_{tr}	Transient feedback gain	pu

Where the time constants T_{g1} and T_{g2} are given by

$$\begin{aligned} T_{g1} &\approx \frac{T_RT_G}{T_G + T_R(g_{st} + g_{tr})} \\ T_{g2} &= \frac{T_G + T_R(g_{st} + g_{tr})}{g_{st}} \end{aligned} \tag{7}$$

following [3]. The governor system gain R and the speed-droop g_{st} are inversely proportional, thus

$$R = \frac{1}{g_{st}}. \tag{8}$$

To run the simulations, typical parameter values are chosen following [3], [6], and [7]. The nominal frequency f_n is set

to 50 Hz. The simulation time is set to 120 s with a step of 10 ms.

The system model is, as seen in Table II, set up with pu parameters. The per unit (pu) system is commonly used in power system analysis, as it is a convenient way to keep a general analysis. The per unit system refers (as a fraction) to a base value. For example the per unit power may be expressed by the fraction $P_{pu} = \frac{P}{P_{base}}$. Thus if the power is given by $P = 1\,pu$, then $P = P_{base}$.

III. Frequency Containment Reserves

The frequency of a power system is dependent on the relation between the electrical power being consumed and produced in the system. The frequency in a electrical power system is constant when the consumed power is equal to the generated power. FCR units objective is to stabilize the system frequency in case of a disturbance as described in [3]. The FCR units are always active, regulating the frequency in the system. For this project the investigated FCR was for large disturbances (FCR-D) which is for contingencies such as the failure of a generating unit or load change [7], which could lead to a large Instantaneous Frequency Deviation (IFD), i.e. the minimum frequency value following a disturbance. If the frequency reaches the maximally allowed deviation, protection systems are triggered. Such protection systems could be UFLS, system separation and steam turbine shut down as stated in [3]. After the disturbance a new Steady-State Frequency Deviation (SSFD) level is reached which differ from the nominal frequency. Deviation from nominal frequency could lead to saturation and power losses if not returned to the original nominal frequency, as eddy currents in the transformers are proportional to the frequency as stated in [8].

An estimation for the SSFD for a linear model following [3] is given by

$$\Delta f_{SSFD} = \frac{-\Delta P_L}{R + D} f_n \tag{9}$$

where Δf_{SSFD} is the steady state frequency deviation, ΔP_L the load change, R the system gain, D the damping constant and f_n the nominal frequency. Hence the main parameters affecting the SSFD after a disturbance is the load disturbance ΔP_L and $R + D$, referred together as the stiffness of the system.

A. Inertia

The inertia in a electrical power system is a prominent parameter that ensures the stability and robustness of the system. The inertia should be sufficiently large so as the IFD does not exceed maximum allowed frequency deviation of the system to avoid triggering protection systems as stated in III.

As stated in section II, the system inertia and the maximal RoCoF are inversely proportional, thus for lower system inertia, a higher IFD can be expected following a large load disturbance. As seen in Fig. 5 the IFD and and RoCoF are visibly larger when the inertia in the system is lower and vice versa. Confirming the theory from section II. Further on, the first and fourth case, the case with highest and lowest inertia respectively, will be the cases focused on for this project. Case 1 will be referred to as the high inertia case and case 4 as the low inertia case.

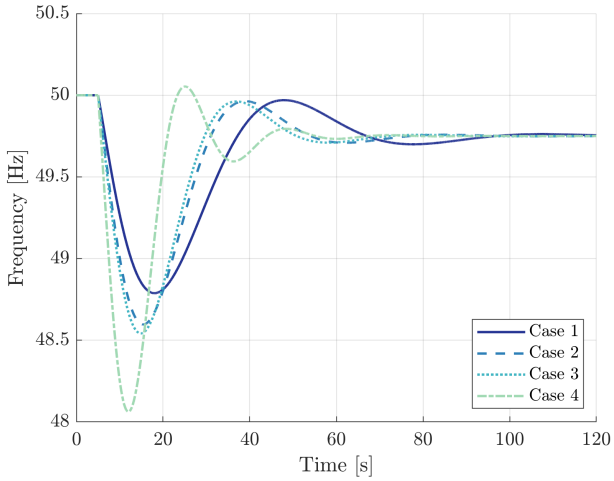


Fig. 5. Frequency response due to a large load disturbance on the four different system inertia cases from Table I, see section II.

B. Hydro

Hydro power generating units have the benefit of being a controllable power source, the amount of power generated can be safely controlled by changing the control gate opening, thus the amount of water flowing through the hydro turbine [3]. The governor-turbine system with frequency as input and mechanical power change as output is modeled as a transfer block, in Fig. 3. The transfer block, Governor, is controlling a turbine gate implying the control of mechanical power output. Hydro turbines however, have a non-minimum phase property as stated in section II which can be observed in Fig. 6. Following a step response, in this case a load disturbance, the initial mechanical power decreases. The non-minimum phase property results in a negative derivative following a step response, thus further extending the power difference ΔP. From the same figure, the synchronous machine dynamics, the swing equation, can be seen using the red dots. When the mechanical and electrical power, thus the generated and the consumed power, are equal, the RoCoF is zero. Furthermore, when the consumed power (electrical) is larger than the produced power (mechanical) the frequency decreases, i.e. the RoCoF is negative and vice versa.

To gain a better understanding of frequency control a sensitivity analysis was done on specific parameters. This was done by altering one system parameter at a time. The analysis will focus on two properties, IFD and SSFD, which are of most interest in FCR study [7]. The analysis is done on the low inertia system, which can be interpreted as a worst case scenario regarding the IFD criteria, see section III-A.

Firstly, variations of the water time constant T_w are performed, see Fig. 7. The changes are done by $\pm 20\,\%$ from the typical value. These alterations show an impact on the

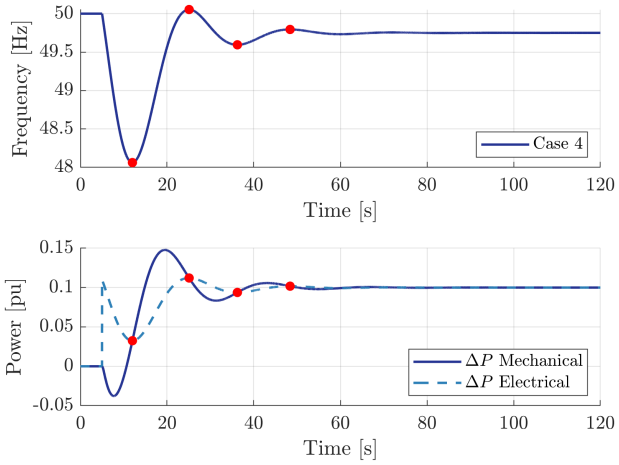


Fig. 6. Frequency response on a low inertia system following a large load disturbance. The red dots are used to indicate the intersection between the mechanical and electrical power, which together with the swing equation (1) are utilized to show the dynamics of synchronous machines.

IFD while no effect occurred to the SSFD. The impact is expected, since a lower time constant results in a faster system response, as the hydro dynamic zeros are moved towards the left half plane, resulting in a more stable system response. Thus the governor has a better chance to counteract the large disturbance.

Variations of the load disturbance ΔP_L are also studied, see Fig. 7, with the parameter changed by $\pm 20\,\%$ from the typical value. As expected from equation (9) an impact on the SSFD is seen. Also a large impact on the IFD is seen. Recall the swing equation (6) a larger load disturbance ΔP_L results in a larger RoCoF considering the other parameters the same. As the system is linearized the frequency response is proportional to the load disturbance ΔP_L, thus the IFD is larger for a larger load disturbance.

Following the same reasoning, an impact on the SSFD is expected when altering the governor system gain R and the damping constant D, see equation (9). This is visible in Fig. 7 for the governor system gain R. There was no visible impact on the IFD due to system gain alterations. Moreover the system gain R is used to ensure a SSFD that is within the accepted ranges set by the Transmission System Operators (TSO). However, a system gain that is too large may result in a unstable system, prone to oscillation due to disturbances. For the damping constant D, the impact on the SSFD was not visible since $R >> D$. However, the damping constant D has an effect on the IFD. Although small, it helps with the response post IFD, minimizing the overshoot and oscillations.

IV. Battery System for Fast Power Support

A. Previous Research

The integration of batteries into power systems for frequency control is an important field of research as the demands on the power systems evolves. For fast frequency support, BESS is a convenient solution as they are both a fast and controllable energy source. Research into BESS are numerous

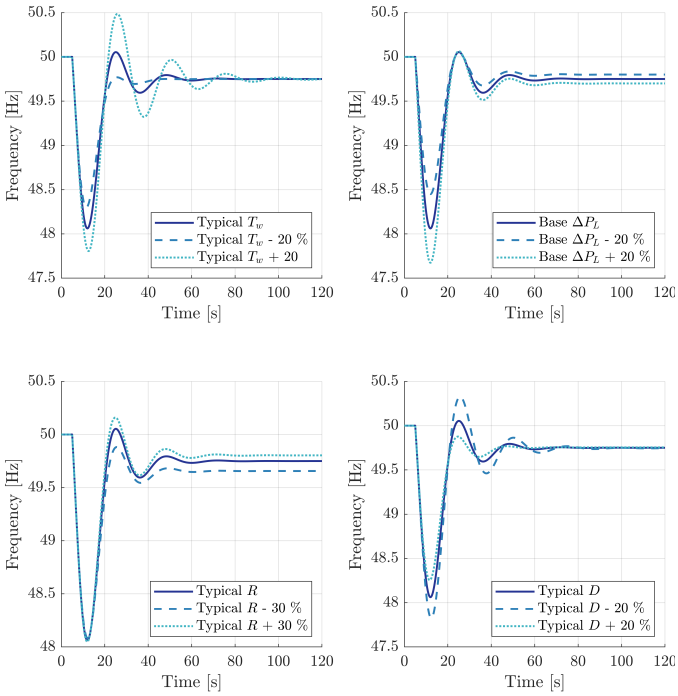


Fig. 7. Alterations on system parameters for a low inertia power system. The upper left plot shows the T_w alterations. The upper right plot shows the ΔP_L alterations. The lower left plot shows the R alterations. The lower right plot shows the D alterations.

and extensive and stretches from using electrical vehicles (EVs) together to form a large battery energy source in [9] to the economics of different battery types used as BESS in [10]. The new demand of renewable energy has made wind and solar power attractive with the cost of a loss of inertia in the power system. In [11] BESS are investigated as a possible solution to the difficulties integrating large amounts of wind power in power systems. As stated in [5], some general guidelines for implementation of batteries in the Nordic system consists of a activation time of the BESS from 49.7 to 49.5 Hz with a maximum full activation time of 1.30 to 0.70 seconds with a support period of 5 to 30 seconds depending on the disturbance.

B. Simulations Including BESS

A model of a BESS was introduced to the Simulink model so that the possible improvement in the frequency response could be investigated. The battery support was assessed and designed for the frequency control during the lowest inertia value, typically presenting the most critical case for frequency control. In the low inertia case, the conventional energy sources (Nuclear) have been completely removed. This means that the only contributing inertia sources are the hydro generators.

The BESS model is activated via a step change. The BESS had a fixed energy capacity E_B of 0.15 pu and a maximum of output power ΔP_{Bmax} of 0.05 pu. The energy was set to a certain value as there are limitations, both economical and physical to the size of the battery and energy that is feasible to

store with modern technology. The triggering level frequency (TLF) is the activation point for the BESS. The triggering frequency is given by

$$f_{trig} = f_n + \text{TLF} \tag{10}$$

hence for a TLF of -1 Hz, the BESS is activated at $f_{trig} = 49$ Hz, for a nominal frequency f_n of 50 Hz. To replicated the dynamics of a BESS a small time constant T_c is introduced to the model. The time constant T_c is set to 50 ms, which is a quite small value considering the dynamics of interest, and therefore provides a near step shaped power response. Thus, for a fixed battery capacity E_B, the total activation time of the BESS is approximately determined for a battery output power P_B by

$$t_B \approx \frac{E_B}{\Delta P_B}. \tag{11}$$

The parameters that were investigated and changed were TLF and P_B with the criteria that the maximum IFD should not deviate more than 48.5 Hz. That is

$$\text{IFD} \geq 48.5\,\text{Hz}. \tag{12}$$

The TLF was iterated from -0.1 Hz to -1.5 Hz with a step of 0.01 Hz and the P_B was iterated from 0.01 pu to 0.05 pu with a step of 0.002 pu. The results can be seen in Fig. 8 which shows the IFD for different TLF and ΔP_B pairs.

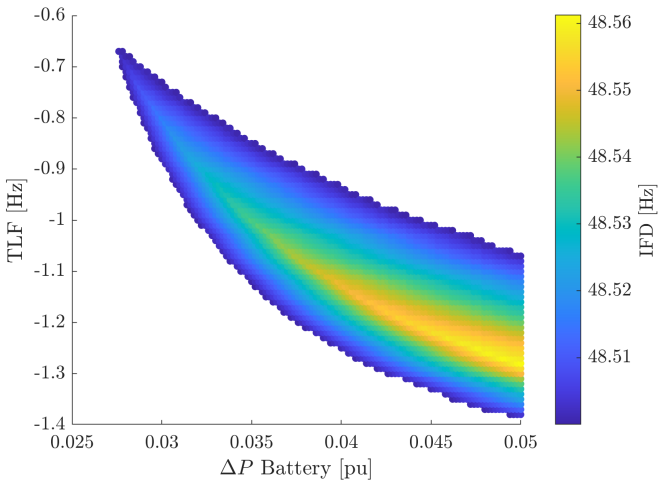


Fig. 8. Scatter plot showing all feasible TLF and ΔP_B pairs fulfilling the IFD criteria of larger than 48.5 Hz.

C. Analysis of Parameter Choice

To achieve the most suited frequency response for our case, the parameter choice of TLF and ΔP_B given some criteria needs to be fulfilled. First the IFD must be higher than 48.5 Hz which all shown parameter pairs in Fig. 8 fulfills. The parameter pairs should also take into consideration the longevity of the BESS. A lower ΔP_B results in lower risk of introducing stress on the BESS, preventing degradation of the batteries due to heat generation from currents as stated in [12]. The last criteria taken into consideration is the avoidance of undesirable activation of the BESS, since this measure is mainly to support the system during severe disturbances which can be prevented with a higher TLF. With this in consideration the parameter pair were chosen to be TLF = -1.15 Hz and ΔP_B = 0.04 pu. This parameter set provided the frequency response which can be seen in Fig. 9 with a IFD of 48.56 Hz. As seen in the figure, the additional battery support successfully aids the FCR units in the low inertia case, reducing the power unbalance momentarily.

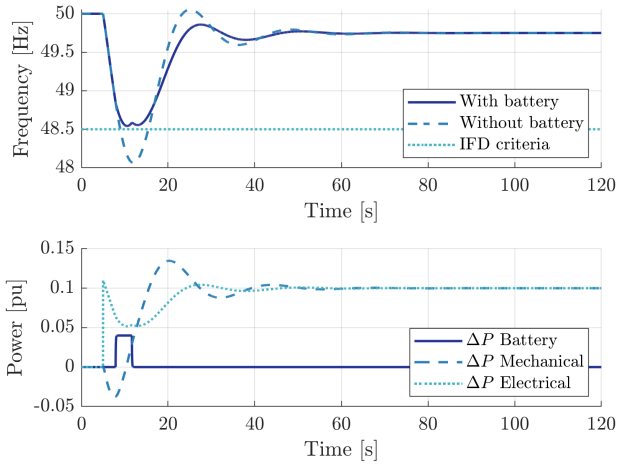


Fig. 9. Frequency response on a low inertia system with a BESS implemented. The battery system triggering level frequency is $TLF = -1.15$ Hz while the injected power is $\Delta P_B = 0.04$ pu.

V. Discussion

A. Our Results in Context

The analysis in section III confirmed the theory provided in section II. The analysis focused on the FCR hydro unit, altering parameters to receive different responses. It was seen that the simulated frequency response confirmed the expected properties of the swing equation. The parameters also had an impact that was expected from the theory. Further, in conclusion it is seen that inertia has a vital part in the frequency stability of a power system. Also, for low inertia systems, the current FCR units may not be enough to guarantee frequency stability. For this reason, a BESS is added to the low inertia system. Which in section IV is shown to improve the frequency response successfully.

This finding is consistent with other studies such as [12], [13] and [14], implementing BESS to improve frequency stability. Where [12] presented a simple controller for utility-scale BESS, showcasing the advantages of said controller reducing the initial frequency drop following a large disturbance. [13] discuss the benefits of BESS inclusion in the power grid, providing synthetic inertia. [14] investigates the effect of BESS installment on grid level transmission grids, showing that BESS can contribute to a change of RoCoF, thus providing a frequency support improving the frequency response.

B. Present Day Methods to Counteract Frequency Deviations

There are present day methods to counteract large frequency disturbances, set to avoid grid scale blackouts. These can be

regarded as short term solutions for a long term problem. Some of those methods are discussed in this subsection.

As discussed in section III-A, the maximal RoCoF and inertia are inversely proportional, thus inserting inertia to a system results in lower RoCoF. Meaning that the FCR units stand a greater chance to counteract the power unbalance. One such method of adding inertia is the introduction of synchronous condensers, for frequency stability reasons. This is investigated in [15] where synchronous condensers where implemented on a Danish grid model, which is mostly wind power based, thus lacking inertia. The results showed that synchronous condensers are reasonable for this application.

Further methods are UFLS, which means that large loads are disconnected after a large disturbance to quickly regain power balance, thus forcing an equilibrium state for the frequency. The implementation of UFLS is discussed in [16] concludes that UFLS is a suitable method to quickly stabilize the system given well designed shedding schemes. The shedding schemes require good knowledge of the system and the disturbance, to quickly determine which loads are to be disconnected. New concepts of load shedding schemes are being developed, one such concept is presented in [17], presenting a stability margin to be considered when determining schemes. The concept is based on the measurement of frequency and RoCoF, combining these measurements to determine a stability margin.

Several methods and concepts are introduced in [5] and [18]. One such is the decrease of output power from the largest units, thus if one such unit trips, the power unbalance is lower, thus the RoCoF is lower. This is a way of dimensioning an accident. Also, starting up already existing, non used gas turbines, as a means of adding inertia ahead of an expected low inertia scenario. However, the use of gas turbines powered from coal, oil, or other non-renewable energy sources would be counterproductive towards the movement of more RES in the system.

C. Future Methods to Counteract Frequency Deviations

An interesting example of a future method to counteract frequency deviations, is using electrical vehicles as energy reserves as a large battery energy storage system for FFR purposes. The amount of inertia varies throughout the day and between the seasons. During the night the total inertia is less than during the day as less inertia contributing loads are connected to the power grid. The same goes for the seasons of the year, winter is generally a high inertia period whilst the summer is a low inertia season as stated in [5]. Fortunately during the night, when inertia is low, cars are also generally parked. Meaning EVs are also parked and charging via the grid. The already established grid connection could be used as an energy reserve. This makes EVs a great possible solution for FFR [9].

Further, as an additional support to traditional FCR units, High Voltage Direct Current (HVDC) interconnections can be implemented as a form of Emergency Power Control (EPC). [19] show that careful design of droop frequency EPC can provide a better frequency response, which can avoid activation of UFLS (see section V-B).

D. Future Research

Future investigation and extension of the project is needed in the area of application on the Nordic synchronous system for FFR, mainly the implementation of BESS in the Nordic system. This would require updating model parameters and adding a complementary BESS that fulfills the requirements of the Nordic system.

VI. Conclusion

The project analyzed frequency control for various case studies of a linearized power system model. The model was used to investigate the impact of inertia on the frequency stability of the system following a large disturbance. This was done by analyzing the frequency response due to a large disturbance, mainly focusing on the IFD and SSFD. The conclusion can be made that inertia has a great impact on the frequency stability of the system, which by analyzing the swing equation, i.e. the main dynamic for synchronous machines, is expected. It could be concluded that less inertia, which is a result of large RES implementation in the power system, results in a worse frequency response considering IFD. A method to compensate for the loss of inertia in the system, was the addition of a BESS for power support. The implementation of BESS as FFR was satisfactory for the given maximum IFD criteria. The analysis of parameter pairs, TLF and ΔP_B, was done with given restraints on the BESS to find the most suitable parameter choices. The parameter pair presented in the paper resulted in a frequency response that meets our criteria for a large disturbance in a low inertia system. In conclusion, BESS was found to be a viable solution in presented case studies to compensate for the loss of inertia in power systems.

Acknowledgment

The authors would like to thank Danilo Obradović for his great guidance and mentorship throughout the project.

References

[1] "Challenges and Opportunities for the Nordic Power System," Statnett, Energinet, Svenska kraftnät and Fingrid, Oslo, Tech. Rep., Aug 2016.
[2] (2020, Jun) Renewable energy explained. U.S. Energy Information Administration, Washington, DC, USA. [Online]. Available: https://www.eia.gov/energyexplained/renewable-sources/
[3] M. Ghandhari, *Stability of Power Systems, An introduction*. KTH Royal Institute of Technology, Stockholm, Sweden: EECS, 2018.
[4] E. Ørum et al., "Future system inertia," ENTSO-E, Tech. Rep., 2015.
[5] (2019, Dec) Fast frequency reserve - solution to the nordic inertia challenge. [Online]. Available: www.fingrid.fi/globalassets/dokumentit/fi/sahkomarkkinat/reservit/fast-frequency-reserve-solution-to-the-nordic-inertia-challenge.pdf
[6] P. Kundur, *Power System Stability and Control*. New York, NY: McGraw-Hill, 1994.
[7] M. Kuivaniemi, N. Modig and R. Eriksson, "FCR-D design of requirements," ENTSO-E, Tech. Rep. 1, Jul 2017.
[8] H. Nee, M. Leksell, S. Östlund, and L. Söder, *Eleffektsystem EJ1200*. KTH Royal Institute of Technology, Stockholm, Sweden: EECS, 2019.
[9] F. Teng, Y. Mu, J. Wu, P. Zeng, and G. Strbac, "Challenges on primary frequency control and potential solution from evs in the future gb electricity system," *Applied Energy*, vol. 194, Jun 2016.
[10] L. Wingren and J. Johnsson, "Battery energy storage systems as an alternative to gas turbines for the fast active disturbance reserve," Master's thesis, Lund University, Lund, Sweden, 2018.

[11] D. D. Banham-Hall, G. A. Taylor, C. A. Smith, and M. R. Irving, "Flow batteries for enhancing wind power integration," *IEEE Transactions on Power Systems*, vol. 27, no. 3, pp. 1690–1697, 2012.
[12] F. M. Gonzalez-Longatt and S. M. Alhejaj, "Enabling inertial response in utility-scale battery energy storage system," in *2016 IEEE Innovative Smart Grid Technologies - Asia (ISGT-Asia)*, 2016, pp. 605–610.
[13] L. Cocchi, C. Pezzato, A. Cerretti, C. Noce, E. Berardinis, and R. Nicolini, "New ancillary services required to electrical storage systems for correct network planning and operation," Jul 2015.
[14] S. M. Alhejaj and F. M. Gonzalez-Longatt, "Investigation on grid-scale bess providing inertial response support," in *2016 IEEE International Conference on Power System Technology (POWERCON)*, 2016, pp. 1–6.
[15] H. T. Nguyen, G. Yang, A. H. Nielsen, and P. H. Jensen, "Frequency stability improvement of low inertia systems using synchronous condensers," in *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2016, pp. 650–655.
[16] Y. R. Omar, I. Z. Abidin, S. Yusof, H. Hashim, and H. A. A. Rashid, "Under frequency load shedding (ufls): Principles and implementation," in *2010 IEEE International Conference on Power and Energy*, 2010, pp. 414–419.
[17] A. Bonetti, J. Zakonjsek, and U. Rudez, "Bringing rocof into spotlight in smart grids: new standardization and ufls method," in *2020 2nd Global Power, Energy and Communication Conference (GPECOM)*, 2020, pp. 238–244.
[18] E. Agneholm et al., "FCR-D design of requirements – phase 2," ENTSO-E, Tech. Rep. 1, Jan 2019.
[19] D. Obradović, "Coordinated frequency control between interconnected ac/dc systems," Ph.D. dissertation, KTH Royal Institute of Technology, Stockholm, Sweden, Oct 2020.

Tuning of Power System Stabilizers to Damp Out Power Oscillations

Joel Viil and Marcus Seisay

***Abstract*—With the rise of global sustainability energy initiatives, the implementation of renewable energy sources in future electrical grids is increasing. Many of the renewable energy sources are however intermittent, meaning they provide varying levels of power. As grids meet the demand of larger loads of intermittent renewable energy sources, small signal instability arises as result of the power oscillations. Small signal instability occurs when a system cannot return to steady state after being exposed to small disturbances. One method to damp power oscillations in an unstable system is by using a Power System Stabilizer (PSS). The goal of this project is to tune a PSS or PSSs required to successfully damp out the power oscillations in a system which is small signal unstable without any PSSs connected. The PSSs are tuned through a trial and error approach, and the system is a Kundur two-area four-machine MATLAB Simulink model. Overall, the trial and error method is successful in tuning PSSs, which damp out the system's power oscillations. Other methods of tuning are discussed and compared in terms of efficiency to damp out power oscillations.**

***Sammanfattning*—Med en ökning av globala hållbarhetsinitiativ förväntas implementeringen av förnybara energikällor öka i elnäten. Förnyelsebara energikällor som sol och vind är intermittenta, vilket innebär att de ger varierande effektnivåer. När nätet belastas med intermittenta energikällor uppstår lågfrekvensfel, vilket skapar oscillationer i spänning. För att dämpa svängningarna i ett instabilt system kan en Power System Stabilizer (PSS) användas. Målet med projektet är att regulera en PSS som dämpar svängningarna i ett system som har lågfrekvensfel. En metod baserat på trial-and-error används för att reglera PSS:en. Detta görs i en Kundur Two-Area four machine System simuleringsmodell i mjukvaruprogrammen Simulink och Matlab. Trial-and-error-metoden lyckas reglera svängningarna med hjälp av två PSS som dämpar effektsvängningarna i systemet. I rapporten diskuteras även alternativa metoder för att dämpa svängningarna i ett instabilt system.**

***Index Terms*—Power System Stabilizer, Critical Generator, Small Signal Instability, Terminal Voltage, Field Voltage, Frequency.**

Supervisors: Angelica Clark. Merhdad Ghandhari

TRITA number: TRITA-EECS-EX-2020:159:

I. Introduction

As the world shifts away from a reliance on fossil fuels for energy needs, nations are instead moving towards renewable energy sources to meet their energy requirements while simultaneously lowering their CO_2 emissions. Although renewable energy sources have no CO_2 emissions, the large scale implementation of them into energy grids creates issues due to their variability. Energy sources like solar and wind are intermittent, meaning they supply varying amounts of power depending on the weather conditions. According to a report done by the Swedish government, the upper limit on the capacity of renewable power production in Spain is capped at 50% of the capacity of the transmission line [1]. This ensures that the total amount of renewable energy sources supplying the grid stays under 50%. This limit is set as past that point grid instability occurs due to power oscillations. The most common result of these oscillations in generator power from renewable energy sources is small signal instability within the electrical grid, which can lead to power outages [2]. There are however different ways to counter these effects and prevent grid instability from occurring. One such control device is a Power System Stabilizer (PSS). A PSS is a feedback controller connected to the generators of a system and which inputs a supplementary signal to counteract the effects that power oscillations can have on the small signal stability of the system [3]. Systems with high gain excitation systems can increase damping torques and help to provide synchronizing power, thereby improving transient stability. However, high gain excitation systems can also degrade this damping torque to the point where the damping torque becomes negative, leading to instability. High gain excitation systems can therefore cause small signal instability as a result of decaying the damping torques [4]. Through the tuning of its parameters, the PSS is able to effectively damp out the power fluctuations caused by intermittent energy sources and prevent large scale outages. The aim of this project is to adequately tune a PSS, which damps out the power oscillations in a system which is small signal unstable Through this project, conclusions can be drawn on how to effectively tune a PSS and how PSSs can be implemented into future electrical grids composed of renewable energy sources. This is done through the use of a Kundur Two-Area System in Simulink, which has all of its PSSs disconnected rendering it small signal unstable [5]. The Kundur Two-Area System is made up of two areas that include eleven buses which are connected by two weak ties [6]. The Kundur Model is often used as a test case for system stability, power interchange, and damping of oscillations [7]. The system is stabilized through the tuning of PSSs on the critical generators, which are responsible for the loss of synchronisation in the system, thereby leading to instability [8]. The PSSs can be tuned through a variety of different methods, although in this project they are tuned through trial and error. At the start the Kundur model has none of its PSSs connected causing it to be small signal unstable. That means that when subjected to small disturbances the voltage oscillations with time grow in the system since they are negatively damped. Small signal instability refers to a system's ability to recover from small disturbances and

once again reach steady-state and synchronism. Synchronism refers to the speed and frequency of the synchronous systems in the network becoming synchronized by reaching the same frequency and thereby becoming stable [9]. Another aspect of system stability is transient stability, which is a systems ability to withstand large disturbances. Usually transient stability is detectable immediately after the disturbance is cleared, with the system exhibiting signs of instability within the first swing or oscillation. The overall aim of the project is to tune the PSSs of the system to adequately damp out the system's power oscillations and have it return to a steady state. We have deemed an adequately tuned system to be one where the settling time is 15 s after a fault occurs. Then through the comparison of other PSS tuning methods a conclusion in regards to whether or not PSSs are a viable option for future electrical power grids and whether their implementation will allow small signal unstable grids to become stable will be made.

II. System Model

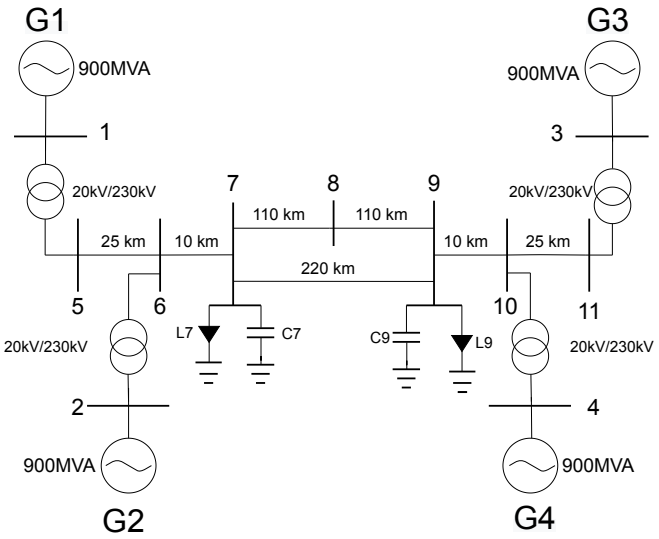


Fig. 1: Kundur Two-Area System.

TABLE I: Power flow table

Generator	P (MW)	Q (MVAr)
1	703	197
2	705	195
3	725	175
4	702	198

The Kundur Two-Area System is made up of two areas that include eleven buses which are connected by two weak ties. The Kundur system used in this project can be seen in Fig. 1. The system has a frequency of 60 Hz, and includes 4 generators, two in each area. Each area includes two generators, each with a rating of 900 MVA connected to a three-phase transformer with a voltage ratio of 20kV/230kV. Table I shows the power flow of the generators in the system when no PSSs are connected. At Bus 7 there is a capacitor with a reactive power of 200 MVAr and a RLC-load with an active power of 967 MW, an inductive reactive power of 100 MVAr, and a capacitive reactive power of 187 MVAr. While at Bus 9 there is a capacitor with a reactive power of 350 MVAr and a RLC-load with an active power of 1767 MW, an inductive reactive power of 100 MVAr, and a capacitive reactive power of 187 MVAr. The capacitor has a capacitive reactive power of 200 MVAr. The load has an active power of 967 MW, an inductive reactive power of 100 MVAr, and a capacitive reactive power of 187 MVAr. Each generator has governor response from a steam turbine and governor. The excitation system used is a IEEE type 1 DC excitation system. At bus 8, a three-phase short circuit to ground fault is placed. This fault does not clear the line and is set with a default switching time of 100 ms from 2-2.1 s. This fault is manipulated throughout the project to aid in identifying the critical generators of the system.

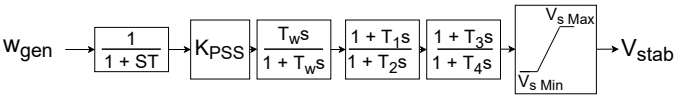


Fig. 2: PSS Block Diagram.

TABLE II: Block Diagram variables and description

Variable	Description
w_{gen}	Generator angle speed
ST	Sensor Time Constant
K_{pss}	Gain constant
T_{ws}	Washout filter
T_1	Lead lag filter numerator
T_2	Lead lag filter denominator
T_3	Lead lag filter numerator
T_4	Lead lag filter denominator
V_{sMax}	Limiter upper value
V_{sMin}	Limiter lower value
V_{stab}	Stabilizing voltage

The system has four PSSs, one for each of its generators. At the start, all of the PSSs are disconnected. The block diagram of the transfer function for a PSS in the system is shown in Fig. 2. The PSS components are described in the Table II. The PSSs have generator speed as their input signal and voltage as its output signal. Of the PSS parameters only five were used to tune the PSS: the two lead lag filters and the gain constant. The lead lag filters were used to tune different ranges of frequencies with the first tuning lower frequencies and the second tuning higher frequencies [10].

III. Methodology

The trial and error method that was used to tune the system included the following steps:

1) Identification of Critical Generator

First the critical generator was identified by increasing the fault time. This was done to bring forth a state of increased

instability to more easily identify the critical generator. The greater a generator's amplitude is when the fault is increased, the more unstable it is considered. The critical generator was hence determined as the generator that participated the most in the unstable mode. This determination was based on two factors: the amplitude of oscillation and the settling time. Therefore, a PSS is most effective when installed at the generator that contributes the most to the overall instability.

2) Tuning of Power System Stabilizer

With the critical generator now identified, the PSS was installed and tuned through trial and error. During the tuning of the PSS, the fault was again shortened, putting the system back into small signal instability. This method required analysis of how a change in each of the parameters impacted the terminal and field voltage as well as the frequency of the generators. Based on the response that was observed with each change, the values were adjusted accordingly.

3) Analysis of System Stability

When the ideal values for each parameter were identified, the system's characteristics were analyzed again. If the system was determined to still be inadequately stable, then an additional PSS or PSSs would be installed until adequate stability was reached. The system stability was evaluated through the application of the fault to assess the margin of improvement by tuning. If a system required additional PSS(s), the next critical generator was identified through repeating the first step with the already tuned PSS(s) installed. Then, the steps of tuning the PSS and evaluating the system stability were executed until the system was able to reach steady state within 15 s after clearing the fault.

IV. Results

The system was able to be tuned following the instructions given in the method. Since the system was small signal unstable at baseline without any PSSs connected, the original fault time of the system was set at a length of 100 ms from 2.0-2.1 s. The small signal instability from the baseline without any PSS connected is illustrated in Fig. 3 and 4, where it can be observed that the oscillations of the terminal voltage and frequency are increasing with time without a PSS connected. The first critical generator was identified by increasing the fault time from 100 ms to 1000 ms. The reasoning behind this was that by increasing the fault time, it would be easier to identify which generator oscillated the most and where to place the first PSS. Judging from Fig. 3 and 4, it is generator 2 that had the greatest amplitude of oscillation. Fig. 5 supports this as when the fault time is increased, it's clear that generator 2 has the biggest impact on system instability. Therefore, placing a PSS there is the most fitting start.

The PSS at generator 2 was tuned through trial and error and was able to successfully damp much of the power oscillations of the system. With the initial values of the PSS that were given in the Kundur model, the strategy was to tune each parameter one by one. The first parameter that was tuned was the gain. The tuning was systematically done through

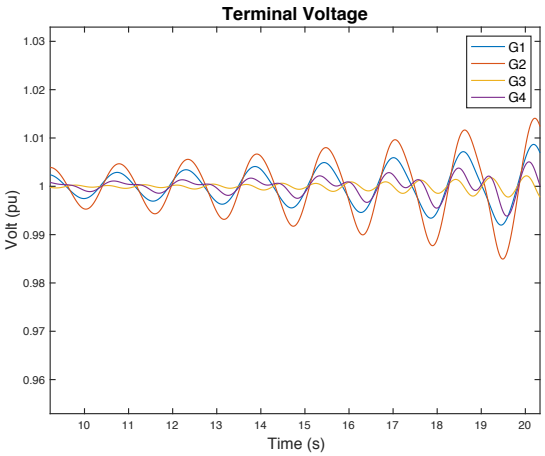


Fig. 3: Small signal Terminal Voltage instability.

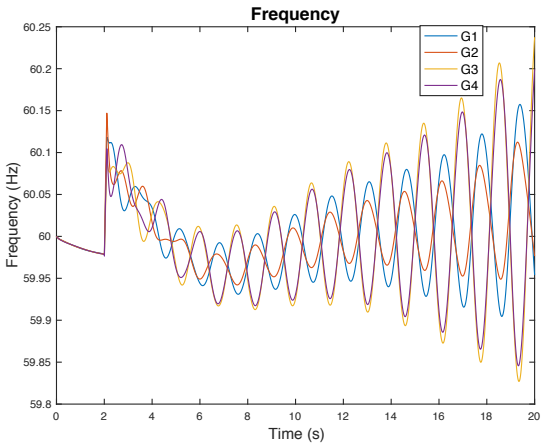


Fig. 4: Small signal Frequency instability.

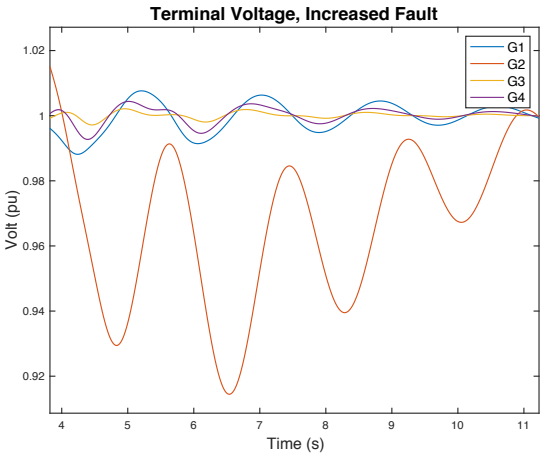


Fig. 5: Small signal Terminal Voltage instability when fault is from 2-3 s.

analysis of the field and terminal voltages and frequency. From the results, a very high gain at first made the system almost reaching steady state in terms of terminal voltage, however over time the system once again became unstable. A very low gain at first made the system amplitude smaller but after a few seconds the oscillations started increasing again, causing the

system to once again become unstable. Then the numerator and denominator of the first lead lag filter were tuned, where the ideal values are based on the ratio between them. However, from experience in tuning the PSS, the first lead lag filter had the biggest impact on the system stability in regards to terminal voltage. When increasing the T_1 to high values of the first lead-lag filter, the oscillations increased a lot. Decreasing this value however, did not make the system reach steady state. When increasing the T_2 of the first lead lag filter to high values, the oscillations of terminal voltage first decreased, but after a short time period, the oscillations are back to increasing making the system more unstable. When decreasing T_2 to low values, the terminal voltages reached steady state earlier. As to the second lead-lag filter, an increase in T_3 makes the terminal voltage unstable and divert from the desired steady state. Decreasing it however, at first contributes to generator 2 becoming stable but then causes it to become unstable again. By increasing T_4 to a very high value, generator 2 first oscillates with a lower amplitude but then after some time the amplitude increases again, becoming unstable. When decreasing T_4 to a very low value, the oscillations increase more than without having a PSS connected, making the PSS counteract its intention. The changes of the values were documented as the PSS on generator 2 was tuned and the ideal values for the PSS of generator 2 is shown in Table III and Fig. 6 and 7.

Gain	Lead-lag 1	Lead-lag 2
4	2 / 0.002	6 / 3

Table III. PSS Generator 2 Final Values

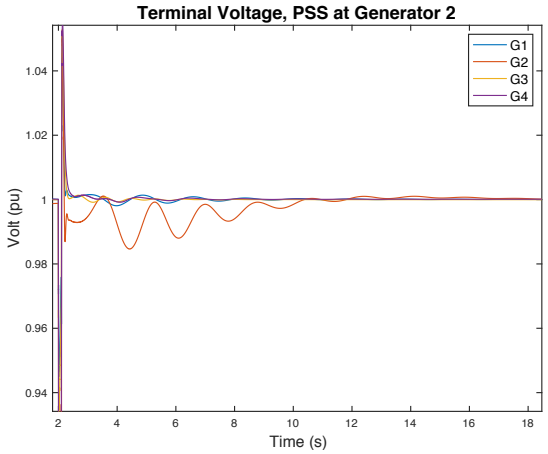


Fig. 6: Simulation Results of Terminal voltage with PSS at generator 2.

Judging from Fig. 6 and 7, the system now returns to steady state, but there is still a need for small improvement of the system's response time. Therefore, after tuning the first PSS, the conclusion was that the system was still inadequately tuned. This was based on although the system now returns to steady state, it still took more than a few swings to become stable, and the settling time was still too large. Therefore, there is still a need for a small improvement of the system

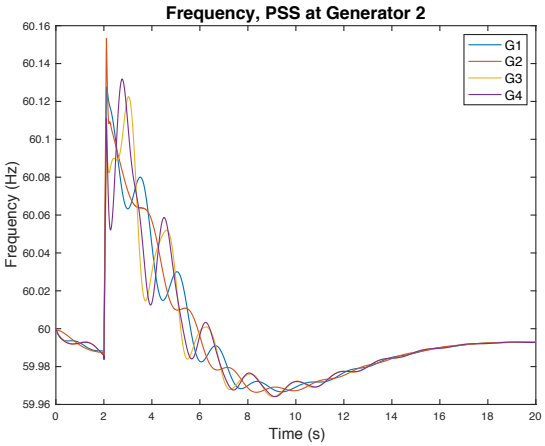


Fig. 7: Simulation Results of Frequency with PSS at Generator 2

response, and finding the second critical generator and tuning it was completed to ensure adequate stability.

The process described in the method was repeated but this time with the first PSS installed. In trying to identify the critical generator, the fault time was again increased from 2-3 s to allow for identification of the second critical generator, while the PSS at generator 2 still connected. The generator that oscillated the most was still generator 2, judging from Fig. 8. But since there was already a PSS installed at this generator, the strategy was to tune the generator with the second largest amplitude of oscillation, generator 1. The methodology of tuning the second PSS was to apply the numbers of the first PSS in generator 2 and then marginally change the values until the results were improved. The ideal values for the PSS of generator 1 is shown in Table IV.

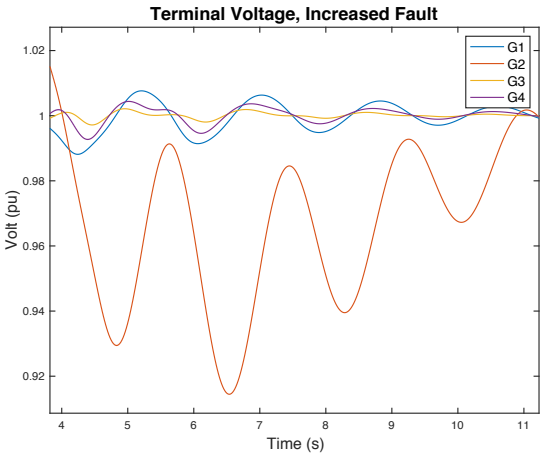


Fig. 8: Simulation Results of Terminal voltage after placing the first PSS at generator 2 and increasing the fault.

Gain	Lead-lag 1	Lead-lag 2
0.1	2.5 / 0.01	3 / 3

Table IV. PSS Generator 1 Final Values

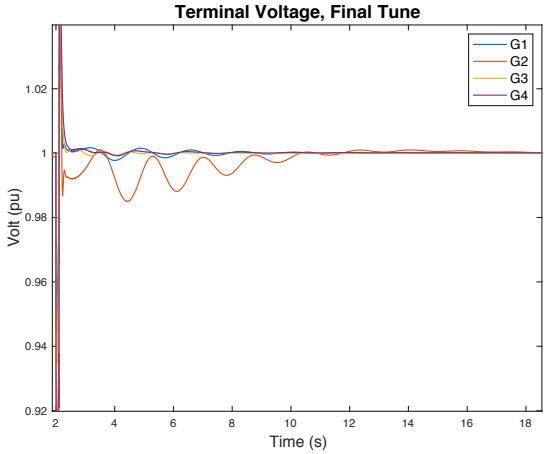


Fig. 9: Simulation Results of Terminal voltage after placing the first PSS at generator 2 and the second at generator 1.

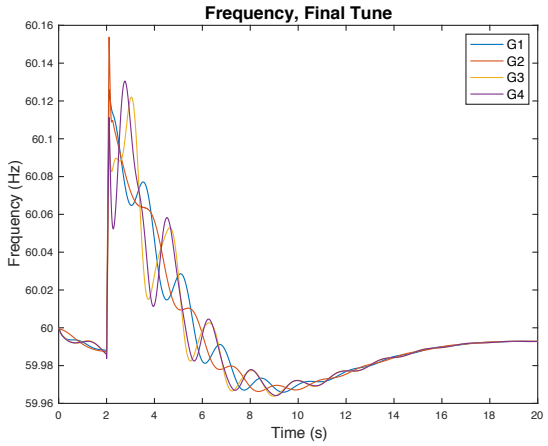


Fig. 10: Simulation Results of Frequency after placing the first PSS at generator 2 and the second at generator 1.

Through the installation and tuning both two PSSs, one at generator 2 and a second at generator 1, the power oscillations of the previously unstable system were able to be damped and the system returned to steady state within 15 s of the fault. There was not a significant difference in system stability with the installation of the second PSS at generator 1.

V. Additional Improvement in Tuning

Although the system's power oscillations were significantly damped with the installation of the two PSSs, the system could be further improved through additional tuning. However, the trial method that was used had the goal of reaching steady state within 15 s from the fault and as the goal was already met, this was not prioritized. The tuning could have been improved either by continuing to tune the parameters in the PSSs that were connected, or by installing additional PSSs at generators 3 and 4. One aspect to keep in mind is that further tuning would be inefficient and wasteful from a resource point of view as the goal was already met.

Another method to damp out the oscillations is to place a second PSS at generator 2. This would be the most efficient since Fig. 8 shows that the oscillations from generator 2 still deviate the most, even after the first PSS was installed. However, the focus of the project was to install a PSS or multiple PSS at different generators and use a systematic trial and error approach to damp out power oscillations in the system. But to further improve tuning, alternative methods are recommended.

VI. Alternative Methods

The trial and error method is not the most optimal method to tune a PSS. This is because the trial and error method lacks the precision of numerical analysis methods. One such analytical method is Eigenvalue Sensitivity Analysis [3]. This method involves linearizing the system, which is often done globally using the built in Model Linearizer Tool in Simulink. Once a global transfer function is obtained, this method then involves finding the eigenvalues to the system's A matrix. After finding the eigenvalues, it is then possible to determine which is the unstable mode within the system. The least stable mode is the one with the lowest damping ratio and frequency. By determining which generator contributes most to the unstable mode, the critical generator can then be identified. The lead lag filter coefficients are calculated from the residue and angle of the most unstable mode. Lastly, K_{PSS} is found by determining the least damping ratio for a range of K_{PSS} values and choosing the maximum value. This is different from the trial and error method used in this study, which found K_{PSS} first and then determined the lead lag filter values [8]. This may have had an impact on the results obtained, especially when considering that the number of lead lag filters used in the trial and error method may not be optimal. Future studies could contribute to overall understanding by using both of the aforementioned methods and comparing them to each other.

PSSs are not the only available systems that can be used to tune unstable electrical systems. Other similar studies have been conducted and have included alternative methods to the PSS. A study was done at Chalmers University of Technology about PSS use in damping power oscillations. This master thesis aimed to create a methodology on how to tune a PSS on a synchronous generator through analysis of the different PSS tuning methods. This study also compares a PSS to a Phasor Power Oscillation Damping (POD) Controller with regards to damping ability. The study concluded that there were many benefits of the Phasor POD Controller over a PSS. This study gives a thorough analysis of the different methods, which can be used to tune a PSS and presents alternative controllers that may be used instead. This shows how although PSSs can effectively damp out power oscillations, there are other alternative methods which could prove to be more effective. In future studies these two methods could be compared but in a Kundur model as opposed to the Sychronous Machine Infinite Bus (SMIB) Model used in the Chalmers report [3].

In real world system stabilization, systems can be built so that they remain stable without the need for stabilizing control systems. A study done by Elsevier analyzed this. This report looked at Photovoltaic impact on system stability, and presented and evaluated different solutions to mitigate this. The conclusion drawn was that system stability is negatively

affected by photovoltaics because they cause oscillations in voltage within the system. This is the same conclusion that serves as the basis for our report. The solution presented in the Elsevier report is one which involves altering the way that electrical vehicles are charged. This provides insight into how grid stability can be approached from a different angle and be prevented instead of being corrected. In this example, a PSS would not be needed as grid stability would already be stable due to its construction [11]. This shows how alternative approaches can be taken to solving system instability.

The trial and error method is effective in tuning a PSS, although other methods may be preferable. This could be due to the fact that the trial and error method is less accurate and may not be as time efficient as other methods, like the Eigenvalue Sensitivity Analysis method. This is due to the reliance of the trial and error method on graphical analysis, which is not as precise when compared to numerical analysis. Other control system could have been more effective than a PSS, which has its own set of drawbacks.

VII. Conclusion

Through the use of a PSS, system stability can be achieved by damping out power oscillations. Although the method used can affect the accuracy and efficacy of the results, the trial and error method is adequate in tuning PSSs to damp out small signal unstable system's power oscillations. The system was able to be tuned through the use of two PSSs, one at generator 2 and the other generator 1. The trial and error approach adopted in this study involved increasing the fault time to help identifying the critical generators. The identification of the critical generators was essential in determining the ideal placement of the PSSs. The parameter values of the gain and lead lag coefficients were determined through graphical analysis of the terminal and field voltages as well as the frequency of the generators of the system. This was effective in tuning two PSSs, which successfully damped out the systems power oscillations.

Other methods such as Eigenvalue Sensitivity Analysis are more effective in certain aspects to the trial and error approach, especially in terms of time needed to tune the PSSs. Through the use of PSSs, future electrical grids can begin to implement more intermittent renewable energy sources without sacrificing stability.

Acknowledgement

The authors would like to thank their supervisor Angelica Clark for her continued support, guidance, and understanding throughout the project.

References

[1] Centeno Lopez, Eva and Ackermann, Thomas, "Grid issues for electricity production based on renewable energy sources in spain, portugal, germany, and united kingdom," p. 63, Jan. 2008. [Online]. Available: http://www.diva-portal.org/smash/get/diva2:467513/FULLTEXT01.pdf

[2] E. Csanyi. (2010, Oct.) Power system stability. [Online]. Available: https://electrical-engineering-portal.com/power-system-stability

[3] Angel Zea, Andrea, "Power system stabilizers for the sychronous generator: Tuning and performance evaluation. master thesis," *Chalmers University of Technology*, p. 60, Jun. 2013. [Online]. Available: https://publications.lib.chalmers.se/records/fulltext/183724/183724.pdf

[4] Gibbard, M.J., Pourbeik, P., and Vowles, D.J., *Small-signal stability, control and dynamic performance of power systems*. University of Adelaide Press, Sep. 2015, Adelaide, AU.

[5] Kundur, Prabha, Balu, Neal J., and Lauby, Mark G., *Power System Stability and Control*. McGraw-Hill Inc., Apr. 1993, palo Alto, CA.

[6] (2021, Apr.) Pmu (pll-based, positive sequence) kundur's two area system. [Online]. Available: https://se.mathworks.com/help/physmod/sps/ug/pmu-pll-based-positive-sequence-kundur-s-two-area-system.html

[7] (2021, Apr.) Two-area sysytem. [Online]. Available: https://electricgrids.engr.tamu.edu/electric-grid-test-cases/two-area-system/

[8] M. Ghandahari, *Stability of Power Systems*. Royal Institute of Technology (KTH), Apr. 2021, Stockholm, SE.

[9] (2021, Apr.) Power system stability. [Online]. Available: https://circuitglobe.com/power-system-stability.html

[10] Glad, Torkel and Ljung, Lennart, *Stability of Power Systems*. Studentlitteratur AB, Sep. 2006, Lund, SE.

[11] Brinkel, N.B.G., Gerritsma, M.K., Al Skaif, T.A., Lampropoulos, I., van Voorden, A.M., Fidder, H.A., and van Sark, W.G.J.H.M., "Impact of rapid pv fluctuations on power quality in the low-voltage grid and mitigation strategies using electric vehicles," *International Journal of Electrical Power Energy Systems*, vol. 118, p. 105741, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0142061519319994

Design of a Future Residential Hybrid Microgrid

Ahmad Talaat Hifzy and Wilhelm Westermark

***Abstract*—As we are moving towards a future carbon-neutral society, development of residential microgrids attracts much attention around the world with its efficient utilization of renewable energy. A residential microgrid is a small power system for a house, which consists of a solar photovoltaic (PV) source, a battery storage, residential loads, and an interface to the grid. In this paper, a hybrid AC-DC microgrid is proposed, studied and simulated in Matlab/Simulink. A coordinated control strategy is developed so that the PV converter is controlled to maximize its power generation, the battery converter is controlled to stabilize the system with the battery state of charge constraints, and an interlinking converter is controlled to decide the connection/disconnection and the power flow with the grid. The simulation results show the effectiveness of the proposed solution under various operating conditions.**

***Sammanfattning*—I det här pappret föreslås, studeras och simuleras ett hybrid-anpassat lokalt självförsörjande elnät i Simulink och Matlab. Solpaneler utgör den distribuerade förnyelsebara energikällan i nätet. Panelerna styrs med en MPPT-algoritm för att maximera kraftgenereringen. Batteriets laddningstillstånd används i det designade batterilagringssystemet för att garantera lång livstid och för att fatta beslut om laddning och urladdning. Kraftöverföring mellan AC- och DC-nätverk sker via en dubbelriktad omvandlare. Det konstruerade hybridnätet fungerar självständigt samt vid sammankoppling till huvudnätet. Ett koordinerat kontrollsystem implementeras för att möjliggöra kommunikationen mellan lokalnätets olika delar. Resultaten från simuleringstestet visar att det föreslagna nätet uppfyller stabilitetskrav och god funktion under varierande driftstillstånd.**

***Index Terms*—Energy management, grid control, grid operation, hybrid microgrid, PV system, MPPT, BESS, DER, BIC**

Supervisors: Qianwen Xu

TRITA number: TRITA-EECS-EX-2021:160

I. Introduction

As a consequence of using conventional fossil fueled power plants to generate power, our planet is heating up dramatically and experts are now certain of the effects of global warming. In addition, fossil fuel will run out at some point. Finding alternative sources of energy a global concern. Renewable energy sources (RES) already attract worldwide attention as a fundamental solution to addressing the power production challenges and environmental threats. Microgrid technology is so far one of the most effective, efficient and promising technologies that can make the conversion to renewable sources possible. This can be done by integrating a distributed energy resource (DER) such as a wind turbine and solar photovoltaics in to the residential power supply chain. According to a proposal from [1], microgrids have been shown to successfully deliver results in economical, technical and ecological aspects, where one of the considerations are that microgrids reduce transmission losses. A DC microgrid can power most of residential appliances, especially since most modern appliances such as electric vehicles and light-emitting diodes (LED) are DC loads by nature. On the other hand, AC microgrids have an advantage when connecting to the utility grid, since there is no need for inverting. As discussed in [2], it is also possible to install a separate AC microgrid in order to support legacy and motor-driven appliance that must be powered by AC. When planning the layout of these two grids, a future expansion should be taken into account and planned for ahead of time, thus avoiding a scenario where brownouts and total lack of power might occur. Planning ahead makes it possible to to supply newer devices whilst not incurring a larger cost. The work from [3] introduces the notion that the grid can operate in both islanded and grid-connected mode. With a grid connection the AC appliances can be easily supplied without worrying about brownouts, as [4] explains. Hybrid microgrids have been attracting the attention recently due to their high efficiency in reducing the stages of power conversion, and success in combining the advantages of both AC and DC microgrids [5] [6].

This paper proposes a hybrid AC-DC microgrid which consists of a PV array for power generation, a power storage system and residential AC and DC loads. The proposed microgrid maximizes power generation from the solar PV panels by employing a maximum power point tracking (MPPT) algorithm. Furthermore, the battery energy storage system (BESS) is designed to balance the PV generation and load with the battery state of charge constraint to guarantee battery lifecycle. An bidirectional interlinking converter (BIC) is designed to regulate the connection/disconnection and the power flow with the grid. Simulations under different operating modes are conducted to show the effectiveness of the proposed solution. The grid maintains stable parameters while operating in both grid connected and islanded mode. The choice of the feasible operating mode is done by considering some essential parameters such as consumption, generation and battery state of charge (SOC).

II. System Design

The system design adopts AC-DC hybrid microgrid topology. It consists of DC and AC subsystems which are linked together by a bidirectional interlinking converter as shown in Figure 1. The DC part of the grid includes the PV system, battery system and DC loads. The AC part includes the grid and AC loads. All the main parts of the proposed microgrid will be explained in this section.

A. PV system

Microgrids are considered to be a key player for integrating distributed energy resources such as photovoltaics . The

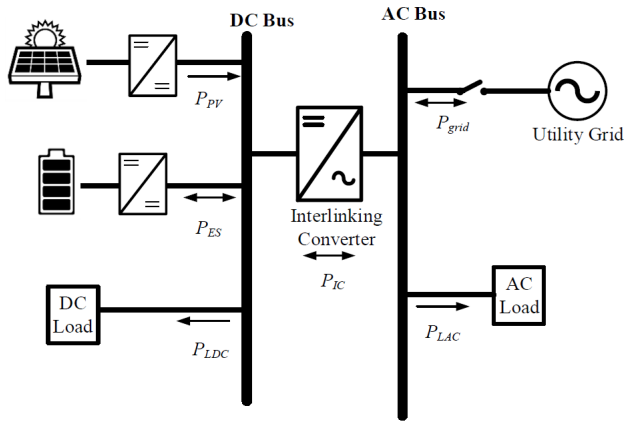


Fig. 1. System topology

energy production of PVs fluctuates depending on environmental conditions. Solar panels are affected by irradiance and temperature, so it is of major interest to ensure that PV is operating at its maximum, despite being deployed in a dynamic environment. This can be done by employing a DC-DC boost converter that operates at maximum power point by using maximum power point tracking algorithm . According to [7] applying MPPT has shown better energy utilization in comparison to directly connected PV systems. A smaller literature study indicated that there are different MPPT algorithms to choose from. An algorithm called Perturb & Observe (P&O) was chosen. It operates by introducing a small perturbation in the duty cycle and observing which way the power and voltage changes. If the power has changed and the differential between the current and previous voltage is either positive or negative, the duty cycle increases or decreases respectively.

B. Battery system

When PVs produce energy, the power it produces has little reaction mass and as the weather changes, so does its output. It is advantageous to store this energy so it can be utilized at a later time and help ease the burden from burst loads. A BESS can be used to bridge the gap. It can be charged with large amounts of power, that can later be used to supply power to a residence. The BESS is connected to the DC bus through a bidirectional converter, so that it can both charge and discharge during microgrid operations. To control the charge/discharge behaviour a two stage control system was constructed, based on a converter design from [8].

C. Bidirectional interlinking converter for grid connection

The differentiating factor of a hybrid compared to an AC or DC microgrid is the communication and the power conversion between the AC and DC networks, which is important in order to supply the loads and transfer the power from the utility grid to the DC side when it's needed. This is why the bidirectional interlinking converter (BIC) is considered to be the most essential part of the hybrid system. The topology of the BIC varies due to the variation of the hybrid microgrid design and configuration. Some topologies adopt three phase AC network [6], and other adopt bipolar DC network [9]. Moreover, new topologies are being proposed [10] [11] to increase the efficiency and reliability in terms of minimizing the power conversion loss and increase the stability. The proposed BIC is designed to operate on a single phase AC voltage, and is based on a two-stage structure. The main component is a bidirectional DC-AC converting stage, and a second stage to reach the required voltage level, since the AC bus has lower voltage than the DC bus. Choosing a single phase AC bus leads to less use of semiconductors devices in the design, which means less cost and smaller volume of the structure [9]. The BIC works as a single phase bridge converter while converting from DC to AC and as a four diode rectifier while converting in the opposite direction. This can be realized by using four semiconductor switching devices. The second stage in the BIC is represented by a transformer to transfer the electrical power between the AC bus and the AC-DC bidirectional converter to produce the desired voltage.

III. Controller design

A. MPPT control

The implemented design features a single boost converter that also includes the MPPT algorithm. Based on the simple configuration that the project intends to simulate Perturb & Observe (P&O) was chosen. P&O is famous for its simplicity and high efficiency. The algorithm is shown in Figure 2.

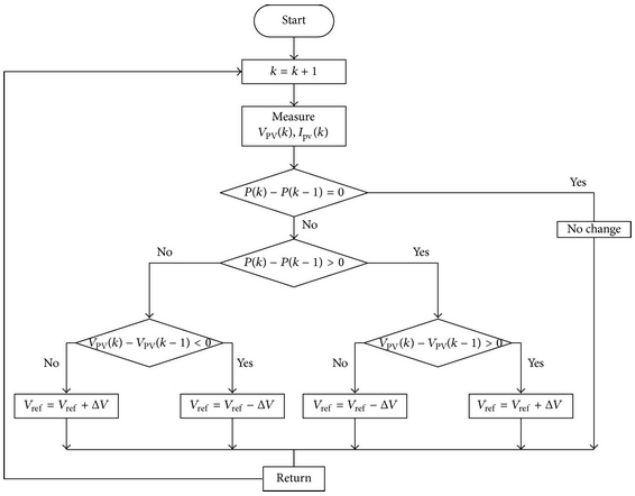


Fig. 2. Flow chart of P&O

Based on the chosen PV array (Zytech Solar ZT320P), the design is capable of producing a theoretical 10253W at an irradiance of $1\mathrm{kW/m^2}$ as seen in the irradiance curve produced by Simulink in Figure 3. With this specification in mind, the P&O algorithm was tuned to perform best at the maximum point at 148V and 79A.

The DC grid was chosen to have a nominal voltage of 760V, which means the MPPT boost converter has a maximum

duty ratio of 0.8053. The P&O algorithm oscillates around this point during normal operation. This value and other components values were chosen based on application notes by Jimmy Hua [12] at Texas Instruments.

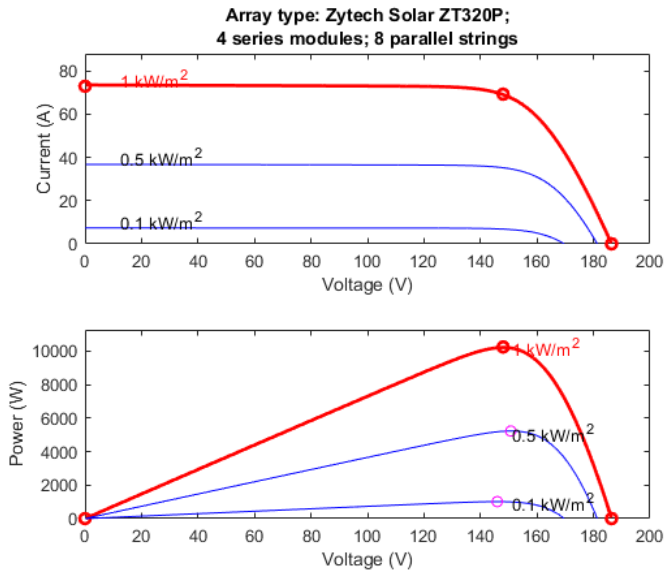


Fig. 3. IV and WV curves with maximum point marked out

B. Battery Energy Storage System with State of Charge-based Control

As more and more storage solutions are deployed, the choice of battery composition will further move towards a Li-Ion based setup [13]. Based on this research a Li-ion battery was chosen as a future-proof technology. A nominal battery voltage of 576V was chosen based on current market designs [14]. The difference in voltage between battery and DC grid makes sure there is a big overhead before the battery is overcharged and allows for high efficiencies when converting between the high and low side. Based on the direction of current flow, the converter design can function both as a buck and boost in both directions, and the double PI-controller from [8] controls the duty cycle for the buck-boost converter that regulates power to and from the battery as shown in Figure 4.

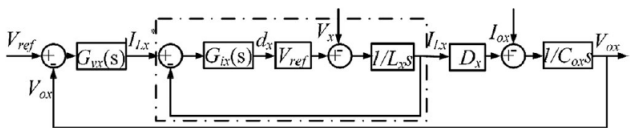


Fig. 4. Control block for the bidirectional DC/DC converter

Because Li-ion batteries cannot handle undercharging or overcharging a Simulink function was devised to stop charge and discharge in case the SOC was out of bounds. In the case where the battery approached 80% charge and still charging, all power to the battery is cut to protect the battery. When the SOC is below 20% a grid connection is established and charges the battery up to 80% charge again.

C. Control of bidirectional interlinking converter

As mentioned earlier, the hybrid microgrid consists of a DC subsystem (PV system, battery system, and DC loads) and AC subsystem (grid and AC loads). Both networks are connected to the BIC which represents the link to the grid. The control of the BIC is essential in order to maintain a stable and functional hybrid microgrid system running under different scenarios. SOC, MPPT, DC bus voltage and AC bus voltage parameters are taken into account while describing the following scenarios:

- Insufficient PV power generation causes a lower voltage on the DC bus than what is required.
- SOC is low and cannot supply the system
- Overloading in the DC network causing DC bus voltage drop
- AC bus voltage variations due to overload in the AC network
- Sudden voltage drop in the grid
- Reactive power that causes instability
- The battery is charged up to 80%

Considering these operating scenarios, the subsequent decisions can be made accordingly:

- Operate in the islanded mode
- Charge the battery
- Discharge through the BESS, allowing the battery to supply the microgrid
- Shift to online mode by connecting to the utility grid

These decisions can be realized thanks to the BIC which makes the power transfer possible. The BIC is designed to invert the AC power to DC while the microgrid is connected to the utility grid. Furthermore, it also converts the DC power to AC when supplying the AC loads in islanded mode. The BICs first stage circuit consists mainly of four switches (insulated-gate bipolar transistor (IGBT)) The inverting process is realized by opening the switches, allowing the diodes to operate and rectify voltage. Thereafter, the output voltage is connected to a large smoothing capacitor in order to flatten the ripple signal. In the other direction the DC-AC converting process is slightly more complicated. The pulse width modulation (PWM) is employed to synthesize the desired AC signal. This is accomplished by switching two IGBTs contrary to the other two.

D. Coordinated Control of Hybrid Microgrid

As previously described, the designed hybrid microgrid operates in two different modes: Islanded and grid connected mode.

1) Islanded Mode: The grid islanded mode implies that the PV array generates enough power to supply both DC loads and AC loads through the BIC. In case of surplus, (PV generation larger than total loads), the battery will be charged. If PV generation is insufficient, battery will support the hybrid system; thus battery serves as an energy buffer, and stabilizes the DC bus voltage at the required value. In this mode, the BIC operates only as a single phase bridge converter to run the AC network.

2) Grid connected Mode: It is not convenient to design a microgrid without the possibility to connect to the utility grid, due to employing the often unreliable DERs as power sources. The proposed microgrid relies mainly on PV and BESS to sustain itself. Consider the case when PV generation is insufficient, and SOC is below 20%. Under-charging a Li-ion battery can both harm the battery and the grid quality, thus switching to the online mode is an urgent need. Connecting to the utility grid priorities charging rather than discharging the battery. Another case is related to the AC bus voltage. The reactive loads affect the AC bus voltage stability and amplitude. Once again, switching to grid connected mode is necessary to maintain the bus voltage within 10% of the specified 220V. This mode allows the utility grid to supply all AC loads and charge the battery. The power transfer is from the AC network to the DC. Consequently, the BICs function in this mode is as an inverter.

IV. Results

The project has had several subgoals, and the results presented below are based upon these subgoals. In the two main modes; grid-isolated islanded mode and grid-connected mode, results are presented from different operating conditions; DC loads, mixed resistive AC, DC loads and mixed resistive and reactive AC load. These aim to show the efficiency of the system, as well as the impact of dynamic loads and changing irradiance. Temperature has been set to 25°C to reduce complexity.

A. Grid Islanded Mode

1) Single DC load:

- Varying irradiance between $250\frac{\text{kW}}{\text{m}^2}$ and $1000\frac{\text{kW}}{\text{m}^2}$
- DC load power at 5kW
- Interlinking converter and grid connection disconnected

As seen in Figure 6 the MPPT boost converter quickly finds an appropriate maximum point when the irradiance changes, and it reaches a maximum of 10106W, which is 147W short of the theoretical maximum extracted from the IV curve in Figure 2. After boosting the voltage to the DC grid level of 760V the efficiency of the MPPT boost converter circuit can be seen in Figure 5. The efficiency is calculated by dividing the PV generated power and the DC load power. It is determined to be between 83% and 88% depending on load and irradiance. A lower irradiance will be more efficient, but produce lower power.

In Figure 7 a 5kW DC load is simulated. Initially the graph of system voltage reads 800V and approximately 5500W indicating that the over-voltage protection diode is dumping excess energy to ground. This protection scheme has been implemented to protect circuitry from burning up and also to help stabilize the system faster. Within 0.2 seconds of system startup, the whole system can be considered stable, as the DC side voltage reaches the target of 760V.

2) Mixed resistive AC and DC loads:

- Fixed irradiance of $1000\frac{\text{kW}}{\text{m}^2}$
- DC load power at 1kW

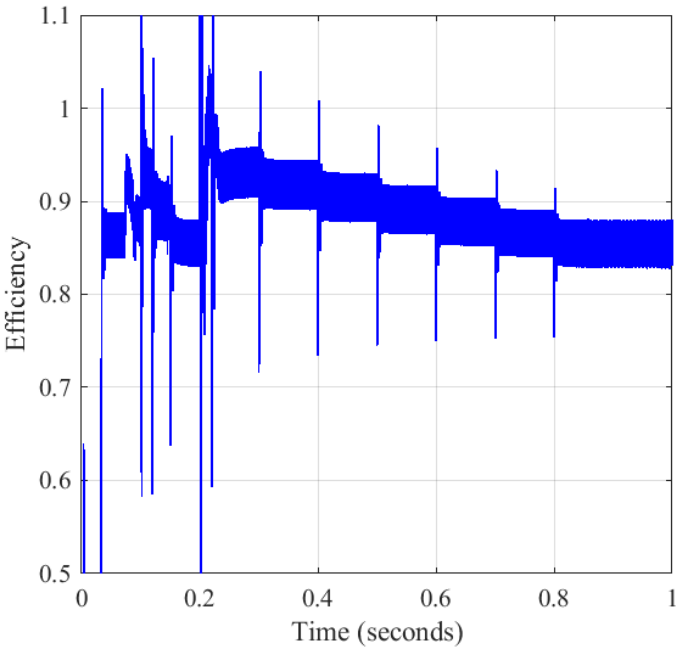


Fig. 5. Efficiency of PV MPPT converter

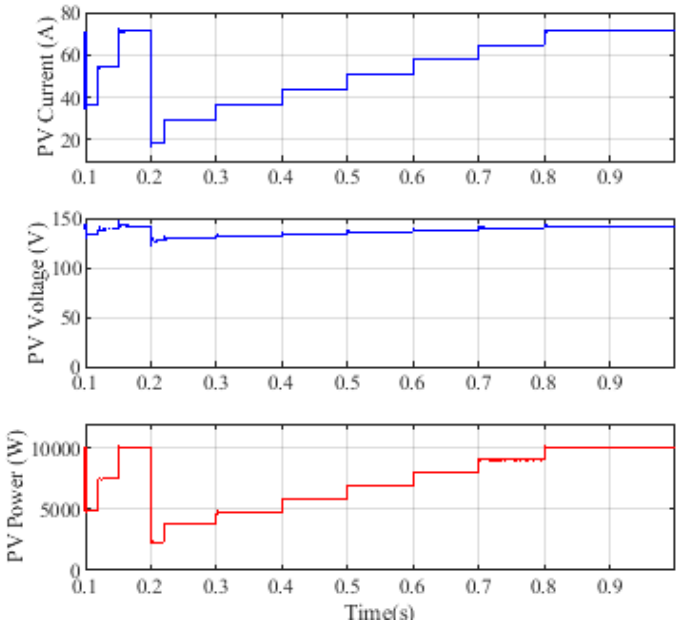


Fig. 6. Statistics from PV MPPT converter

- AC load power at 4kW, 6kW after 0.25s

With the BIC connected, an AC load can now be supplied power from the DC side. Ignoring some initial conditions, the resulting graph in Figure 8 indicates that the BIC provides a stable voltage, peaking at 218V, placing it well within the 10% limit mandated by EU standard EN50160 [15]. Some variance is to be expected when using a transformer with fixed windings. After switching to a 6kW, the voltage drops to 200V, still within standards. The BIC also introduces a slight oscillation on DC grid voltage, seen in Figure 9, this

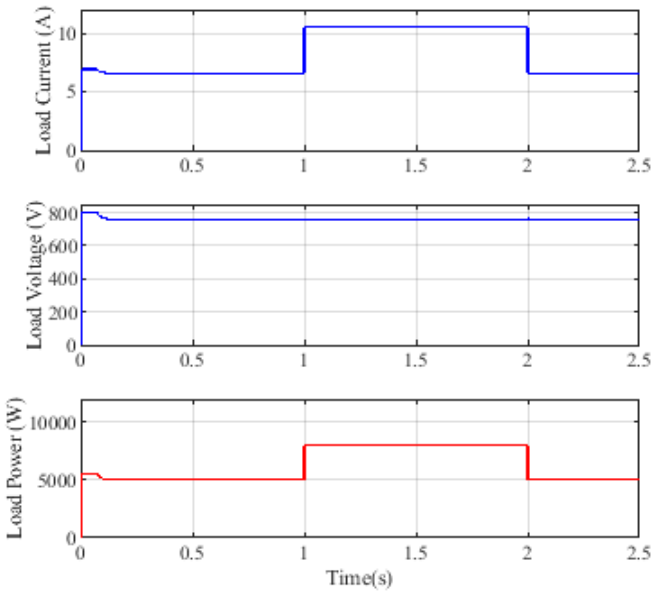


Fig. 7. Statistics from DC load

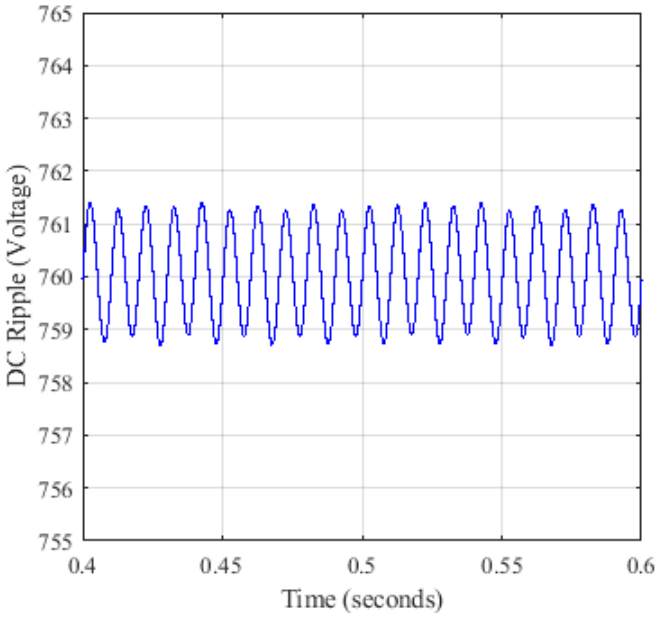


Fig. 9. Excerpt from the DC load with IC connected

behaviour also introduce more noise in the MPPT controller. This in turn increases the spread of the efficiency to between 80% and 90% compared to the non-BIC case of $83\% - 88\%$ seen in Figure 5.

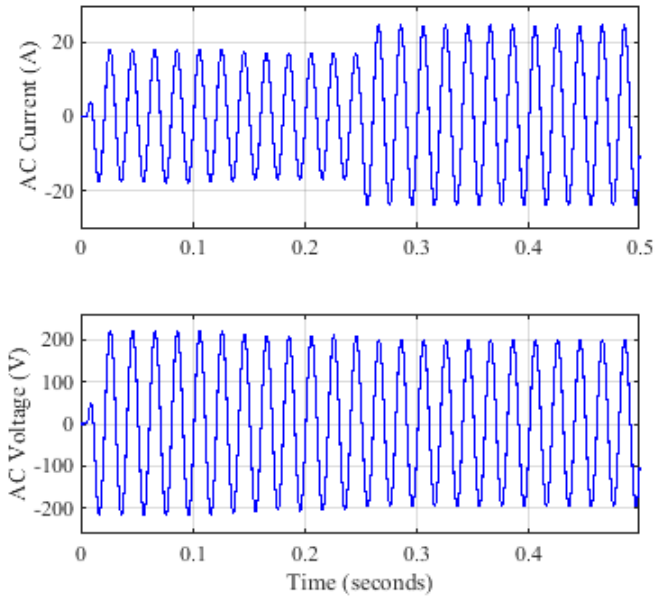


Fig. 8. Statistics from AC load

B. Grid connected Mode

1) Mixed resistive and reactive AC load:

- Fixed irradiance of $1000\frac{\text{kW}}{\text{m}^2}$
- DC load power at 0W
- AC load power at 5kW
- AC reactive power at 1kV A (negative var)
- Interlinking converter and grid connection connected after 0.2 seconds

As seen in Figure 10 the microgrid can handle large residential AC loads. After initial turn-up, the system correctly assumes the islanded mode. Figure 11 shows the moving average function finding a good approximation of the AC voltages peak-to-peak maxima, which is one deciding factor between islanded mode and grid connected mode:

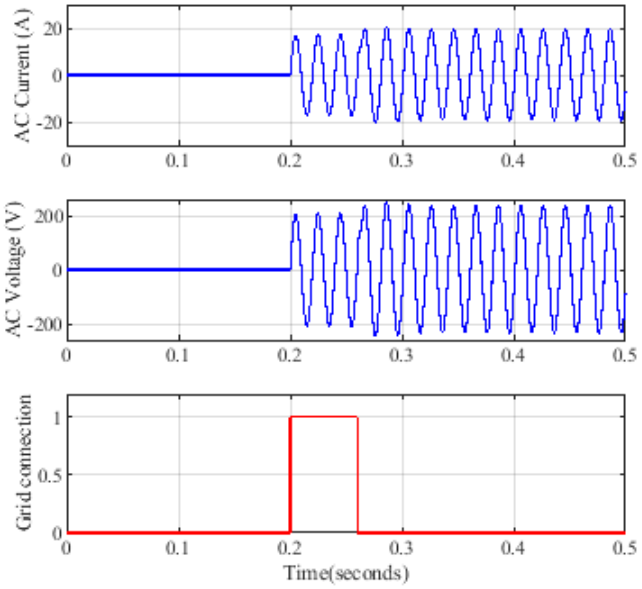


Fig. 10. Stable AC voltage after disconnect from grid

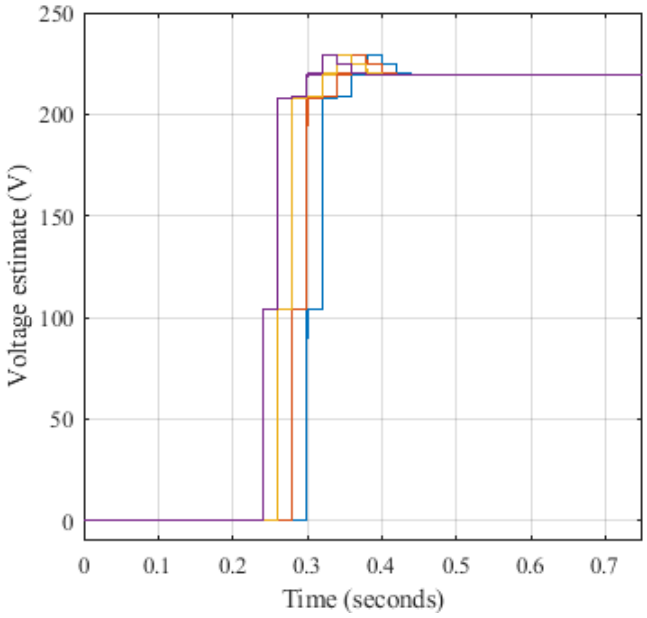


Fig. 11. Averaging function to find peak-to-peak maxima

- Fixed irradiance of $1000\frac{\text{kW}}{\text{m}^2}$
- DC load power at 0W
- AC load power at 5kW
- AC reactive load of 2.5kV A (positive var)
- Interlinking converter and grid connection connected after 0.2 seconds

Due to a high reactive load, the configuration shown in Figure 12 is unable to produce an acceptable grid voltage, stabilizing at $V_{rms} = 252V$ and the coordinated control scheme decides that a grid connection is needed to stabilize the AC microgrid. The scheme will adjust to the new voltage and try to disconnect. If this occurs three times, the control scheme acknowledges that the reactive load is too large, and will engage the grid connection on a more permanent basis.

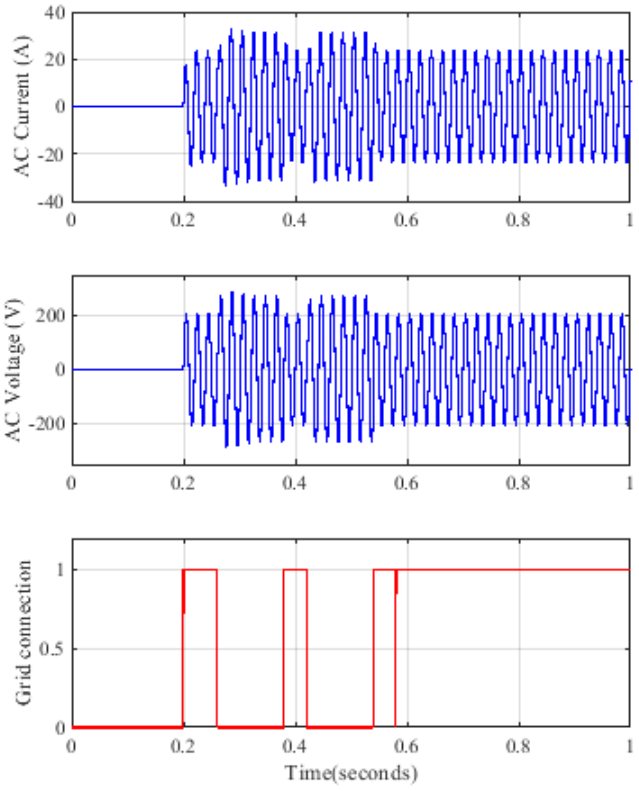


Fig. 12. Grid voltage fallback after three cycles

V. Discussion

Designing a hybrid microgrid that relies on the PV solar power as the main energy source is challenging. Therefore, it is important to guarantee that the PV operates at is maximum. Thanks to MPPT algorithms, this can be fulfilled. Among other MPPT algorithms such as incremental conductance or constant voltage, the choice fell on Perturb & Observe. Because of its high efficiency, fast response and simplicity, since it requires neither advanced knowledge in the PV nor measurement of individual solar cell temperatures [16]. The P&O algorithm is implemented by using different control methods. Control by duty ratio is the chosen method in this design. Other control methods could also be used in the implementation. According to [17] the current control method would have the same functionality and efficiency.

Regarding the BIC. Semiconductor switching devices are fundamental in the inverting and converting processes. An acceptable BIC design could also be obtained by using MOSFET. IGBTs often has more advantages than MOSFET as stated in [18]. IGBT can in general supply power more efficiently when connecting to a DER. Furthermore, IGBT provides better sinusoidal voltage waveforms.

The second stage of the implemented BIC is able to transform the voltage from the utility grid into the BICs first stage and vice versa. This part could be completed differently by modifying the the voltages between the DC bus and the BICs first stage. As a consequence, obtaining 220V bidirectional conversion instead of the 760V designed. This is possible by implementing a buck-boost DC-DC bidirectional converter. Despite this, the transformer is more desirable, due to its design simplicity and stabilizing ability in the AC network. The transformer is however not able to handle large amounts of reactive loads, as that skews the AC voltage out of bounds. Figure 12 shows the phenomena well.

To provide a good estimation of AC voltage peaks over time, the project members employed a sampling function that outputs a moving average of the AC voltage peak. It performs well under normal loads, and converges quickly, as seen in Figure 11. Under abnormal loads, where the grid connection is needed, the algorithm might not converge, thus creating the need for a permanent grid connection for the remainder of the simulation. Another solution would be to aggregate all loads and calculate a maximum load. The coordinated control scheme will then activate the grid for all instances when the load is too big. Figure 13 shows the moving average not converging below 242V within three attempts of grid

connections. The grid engages between 0.6s and 0.7s

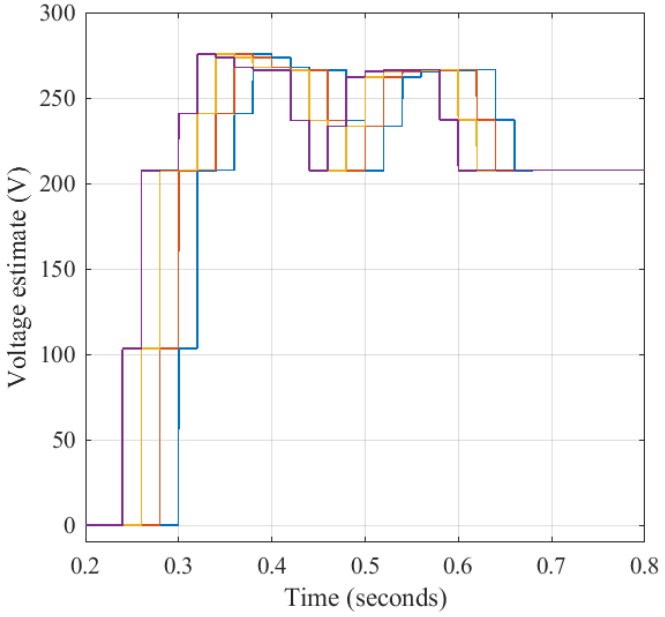


Fig. 13. Unstable moving average with grid connection

VI. Conclusion and Future Work

In this paper a hybrid AC-DC microgrid is proposed, studied and simulated in Simulink and Matlab. PV panels represent the DER in the grid, and it is controlled to maximize its power generations by employing a P&O MPPT algorithm. SOC is being estimated in the designed BESS in order to guarantee long battery life time, and make decisions regarding charging and discharging. The power transferring between both AC and DC networks is done through the BIC. The proposed microgrid operates in islanded and grid connected mode. Furthermore, coordinated control system and strategies are implemented in order to maintain good communication between the grid parts, and ensure acceptable functionality and stability due to various operating conditions. Different reliable scenarios are proposed to test the designed grid. The simulation test results show that the BIC transformer can not manage large reactive loads. A variable winding transformer and switching code would be one way of handling the change in peak-to-peak voltage, induced by the reactance. The microgrid also suffer from transients which appear initially when connecting to the utility grid by the BIC. Those transients are not desirable in the real world, and future work in this area include handling these transients by slowly phasing in the BIC instead of instantly switching it on. Future projects can consider these transient more carefully, and optimize the grid functionality towards higher efficiencies. In addition, there is still room for improvements and further modifications. For instance, spot pricing dependent charging by using data from Nord Pool. which can be useful to make decisions such as selling power back to the system when the price is high, or charging the battery when the price is low. The coordinated control scheme can also benefit from better safety features. Instead of turning off the battery connection when the SOC is above 80%, it could switch from MPPT, and match the load instead of maximizing it. Currently the control schemes under-voltage protection only puts the system into emergency charge mode, it could be improved to allow different ways of charging the battery.

Appendix A
Printout of hybrid microgrid Simulink system

Appendix B
Printout of MATLAB code for Simulink model

Acknowledgment

The authors would like to thank Linus Dahlgren, E-12 for his support. We would also like to thank Qianwen Xu for her guidance during the project.

References

[1] N. Savic, V. Katic, B. Dumnic, D. Milicevic, Z. Corba, and N. Katic, "Cost-benefit analysis of the application of a distributed energy sources in the university campus microgrid proposal," in *IEEE EUROCON 2019 -18th International Conference on Smart Technologies*, 2019, pp. 1–6.

[2] Yasmin Ali. (2020, Oct) Research explores dc lighting and building microgrids market. Westborough, MA 01581. [Online]. Available: https://www.world-energy.org/article/13070.html

[3] H. Lotfi and A. Khodaei, "Ac versus dc microgrid planning," *IEEE transactions on smart grid*, vol. 8, no. 1, pp. 296–304, Aug 2015.

[4] N. M. Tabatabaei, E. Kabalci, and N. Bizon, *Microgrid Architectures, Control and Protection Methods*, 1st ed., ser. Power Systems, 2020.

[5] F. Gao, X. Wang, P. Yang, S. Kou, and M. Sun, "Research and simulation of hybrid ac/dc microgrid," in *2020 4th International Conference on HVDC (HVDC)*, 2020, pp. 1276–1280.

[6] T. Ma, M. H. Cintuglu, and O. Mohammed, "Control of hybrid ac/dc microgrid involving energy storage, renewable energy and pulsed loads," in *2015 IEEE Industry Applications Society Annual Meeting*, 2015, pp. 1–8.

[7] M. A. Elgendy, B. Zahawi, and D. J. Atkinson, "Comparison of directly connected and constant voltage controlled photovoltaic pumping systems," *IEEE Transactions on Sustainable Energy*, vol. 1, no. 3, pp. 184–192, 2010.

[8] Q. Xu, X. Hu, P. Wang, J. Xiao, P. Tu, C. Wen, and M. Y. Lee, "A decentralized dynamic power sharing strategy for hybrid energy storage system in autonomous DC microgrid," *IEEE Trans. Ind. Electron.*, vol. 64, no. 7, pp. 5930–5941, 2017. [Online]. Available: https://doi.org/10.1109/TIE.2016.2608880

[9] P. Najafi, A. H. Viki, and M. Shahparasti, "Evaluation of feasible interlinking converters in a bipolar hybrid microgrid," *Journal of Modern Power Systems and Clean Energy*, vol. 8, no. 2, pp. 305–314, 2020.

[10] E. F. Vidal and I. Barbi, "Ac-dc bidirectional single-phase step-down converter with high power factor," in *IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics*, 2006, pp. 2043–2048.

[11] H. Qian, J. Zhang, J.-S. Lai, and W. Yu, "A high-efficiency grid-tied battery energy storage system," *Power Electronics, IEEE Transactions on*, vol. 26, pp. 886 – 896, 04 2011.

[12] Jimmy Hua. (2019, Apr) Output noise filtering for dc/dc power modules. Texas Instruments, Post Office Box 655303, Dallas, Texas 75265. [Online]. Available: https://www.ti.com/lit/an/snva871/snva871.pdf?ts=1619425061783

[13] P. Larsson and P. Börjesson, "Cost models for battery energy storage systems," Bachelor's Thesis, KTH Royal Institute of Technology, Stockholm, 2018. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-235914

[14] Ferroamp. (2020, Mar) Pylontech energy storage. Ferroamp Elektronik AB, Domnarvsgatan 16, 16353 Spånga, Sweden. [Online]. Available: https://ferroamp.com/download/pylontech-h1-datasheet

[15] Antoni Klajn, Marta Batkiewicz-Pantula. (2017, Mar) Voltage characteristics of grid electricity (en 50160). Leonardo Energy, https://leonardy-energy.org. [Online]. Available: https://www.slideshare.net/sustenergy/voltage-characteristics-of-grid-electricity-en-50160

[16] M. A. Elgendy, B. Zahawi, and D. J. Atkinson, "Operating characteristics of the p o algorithm at high perturbation frequencies for standalone pv systems," *IEEE Transactions on Energy Conversion*, vol. 30, no. 1, pp. 189–198, 2015.

[17] J. Lee, J. Jo, and H. Cha, "Mppt performance comparison between duty-cycle control and current control for photovoltaic power conditioning system," in *2018 21st International Conference on Electrical Machines and Systems (ICEMS)*, 2018, pp. 1036–1040.

[18] C. Natesan, A. Devendiran, S. Chozhavendhan, D. Thaniga, and R. Revathi, "Igbt and mosfet: A comparative study of power electronics inverter topology in distributed generation," in *2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]*, 2015, pp. 1–5.

CONTEXT G
FUSION – THE SUN'S ENERGY SOURCE ON EARTH

POPULAR DESCRIPTION

Unlimited Power

Fusion is the power of the Sun. Harnessing it on Earth would pave the way to long term sustainable energy. Fusion power can potentially yield unlimited energy. Thus, your home will be powered by the same energy as is produced in the center of the Sun.

Most of the world's energy is produced by fossil fuels. Fossil fuels emit greenhouse gases. This changes the climate on Earth leading to more unpredictable weather and rising seas. Fusion power is completely free from those emissions and produces more than a million times the energy than for example burning coal.

There are many difficult problems that need to be solved before fusion can become a reality. Extremely hot gas, called plasma, needs to be heated to more than 100 million degrees centigrade. It has to be contained with powerful magnetic fields and produce more energy than consumed. These conditions are very expensive to create and difficult to control, which is why fusion is not a common source of power in modern society.

Today, nuclear power plants run on fission power. Fission is the process of colliding a neutron into a large atom, usually uranium or plutonium, splitting it into two smaller atoms, while fusion occurs when two atoms, such as two hydrogen atoms, collide together and form a heavier atom, helium. Fusion is the main process that powers the Sun and produces several times the energy than that of fission.

ITER ("The Way" in Latin), which is a fusion reactor currently being built in the south of France. It will attempt to produce 10 times more energy than put in and will start its operations in 2025. Hopefully, this will prove that fusion is a viable energy source. If fusion becomes a success, you will gain access to unlimited power.

SUMMARY OF PROJECT RESULTS

Fusion is the process that happens in the sun and other stars and typically involves two lighter elements fusing together. In this process there is a difference in mass in the end product which reflects how much energy was released in the process according to $\Delta E = \Delta m c^2$.

There have been attempts to replicate this on earth for a long time, as there is a lot of energy released and the process is relatively "clean". Fusion happens in the core of the sun where the pressure and heat is enormous. On earth, most of the attempts involve using very hot plasma, which needs to be heated to at least 100 million K.

As no material can withstand this amount of heat, the plasma needs to be contained in some way. The most well studied way of containing the plasma is using a Tokamak reactor, which contains the plasma using magnetic fields. This is possible due to the conductivity of the plasma. The most energy efficient fusion reaction is between the hydrogen isotopes deuterium (D) and tritium (T) according to

$D + T \rightarrow He + n + 17.6MeV$.

The benefits of using fusion to generate energy is that the process is free from CO_2 and is extremely energy dense. Due to the small amount of fuel needed to operate, there is no chance of a runaway reaction, which could be dangerous due to the rapid increase in temperature and pressure. This could lead to an explosion. The fusion process does produce some radioactive waste due to the neutrons released but the radioactivity of the waste is relatively short lived. There are attempts to use the

neutrons to produce tritium by lining the inside of the reactors with lithium, which when bombarded with neutrons decays into tritium. Deuterium is a fairly abundant isotope and can be extracted from sea water.

In project G1, we implemented a method, using python, to reduce the noise of the ion beam analysis technique, Elastic Recoil Detection Analysis (ERDA). This method predicts the energy to time of flight distribution of specific elements found in the wall-materials of Tokamak reactors. This was done by producing a simulation of the recoiled ions and a noise distribution caused by the time of flight detectors with an older set of data from wall-materials.

In project G3, our purpose was to study dispersion in JET tokamak plasma in conjunction with radio wave heating. The analysis has been done using the Matlab based code FEMIC (Finite Element Model for Ion Cyclotron heating), which renders a COMSOL model file where simulations of the plasma have been studied.

The dispersive effects have been examined through choosing certain parameters such as toroidal mode number, partial power absorption of ions and centering of power absorption. Through these parameters, a quantitative analysis of plasma dispersion has been possible.

The dispersion relation, which relates the refractive index to the wave number in the plasma shows a singularity. This singularity has been studied in relation to our chosen parameters. Lastly, a 3D electric field was plotted using simulated 2D field data.

Our analysis shows that dispersion has a strong dependence on toroidal mode numbers where dispersive effects are prominent in low toroidal mode numbers and decreases steadily for increasing mode numbers. We have found that the singularity in the dispersion relation does not significantly affect the power absorption for mode numbers greater or equal to 8. In toroidal mode numbers less than 8, we have not been able to render a FEM model with sufficient resolution to do a meaningful analysis.

The materials used in **project G1** comes from the JET reactor in the United Kingdom. The JET, much like the ITER, is a Tokamak-type reactor. Therefore, the results from this project will contribute to the improvements of the ITER and other tokamaks, which will add to the advancement of fusion power.

In project G3, we found that the singularity did not have a big impact on the results of the simulations, which means that the FEMIC code does not make a big error due to the dispersion relation in means to power absorption.

In future projects similar to **G1**, more advanced methods could be used to find a more accurate prediction of the distribution by using machine learning algorithms. This would yield more concrete results, which could be applicable to larger sets of data.

In project **G3**, one could also simulate with better computational power at higher resolution to find more accurate results at lower toroidal modes. Another scenario could also be different plasma configurations, for example with tritium added or different isotopes.

IMPACT ON SOCIETY AND ENVIRONMENT

Fusion power could affect the individual and society as a whole in many ways. One aspect would be the increase in energy consumption since fusion is virtually an unlimited energy source. This would decrease the cost of energy and would in consequence increase the standard of living for the consumers. This would in turn increase the consumption of other goods and services in society, contributing to the overall economic cycle, hence promoting growth in other industries.

The downsides of fusion power would be the layoff of the workers in other energy sectors, such as the oil and gas industry, while working side by side with renewable energies like solar and wind power. This could potentially cause a high unemployment rate during a transition period. A solution to this problem could be to keep the transition under government regulations and charge for it temporarily but at the same time slowly phase out the rest of the energy industry and make it as cheap as possible for the consumers. The increased economic activity in society would most likely promote more jobs in

the long run. Such a solution could reduce short run and long run unemployment, which would further improve the standard of living.

Another disadvantage would be the high initial costs of implementing a fusion reactor. Therefore, only countries who could afford it would be able to benefit from it, which means mainly industrialized countries would benefit from fusion technology. This would be unfair since 3rd world countries would lag behind in the technological development of the energy sector.

The biggest impact of fusion power will likely be on the environment. As the energy production is free from CO_2, it could replace other CO_2 intensive energy production methods such as coal. This would counteract the problems connected with it, such as climate change. Fusion power will likely be a base power, meaning it will not likely replace on-demand and regulative energy sources such as hydro power but more constant ones such as coal.

There are some radioactive materials created in the process due to the neutrons released, which will get caught in the material surrounding the reactor making it radioactive. Radioactive material is dangerous to our health since it can kill cells and cause mutations to our DNA, which can cause skin burn, hair loss and increased risk of cancer. There are, however, experiments where researchers try to use the neutrons to create tritium, an essential part of the fusion process by lining the reactor with lithium. The goal with this is to absorb 95% of the neutrons and create more tritium. The 5% that does not get absorbed by the lithium however needs to be handled. The material will be radioactive with life-times of about 100 years, which is better than for example fission power that creates radioactive material with life-times of about 100 000 years.

Due to the quantity of neutrons released, there is also a risk for a fusion power plant to be used to produce material for nuclear weapons by lining the reactor with material to enrich into weapons grade uranium and plutonium. However, it would not be economical to use a fusion power plant to make such material and much easier to use a fission power plant for that.

Overall, fusion has a net positive impact on the world and is, therefore, very valuable to our future. We dream that one day fusion will be available for commercial use, so let's hope that it will soon be up and running.

Data Processing in Accelerator-Based Analysis of Wall Materials From Controlled Fusion Devices

Arvin Quoreshi

Abstract—**The goal for this project was to analyze and understand the noise of the ion beam analysis technique, Elastic Recoil Detection Analysis (ERDA). This was done by examining two models: Classical models and a prediction model. The prediction model is a parameterized noise distribution model. After examining the models, we concluded that both models had advantages and disadvantages for ERDA analysis. This information could be applicable to our understanding of how ERDA could improve for our analysis of wall materials, which could lead to the overall development of fusion reactors.**

Sammanfattning — **Målet för detta projekt var att analysera och förstå brus från jonstråleanalystekniken, Elastic Recoil Detection Analysis (ERDA). Detta gjordes genom att granska två modeller: Klassiska brusreduceringsmodeller och en förutsägelsemodell. Förutsägelsemodellen är en parametrerad brusfördelningsmodell. Efter granskningen av modellerna drog vi slutsatsen att båda modellerna hade fördelar och nackdelar för ERDA-analys. Denna information kan vara tillämplig på vår förståelse av hur ERDA kan förbättras för vår analys av väggmaterial, vilket kan leda till den övergripande utvecklingen av fusionsreaktorer.**

Index Terms— **ERDA, ToF, Ion Beam Analysis, Potku**

Supervisors: Laura Dittrich, Per Petersson

TRITA number: TRITA-EECS-EX-2021:161

I. Introduction

THE interaction of the plasma with surrounding wall-materials in a fusion reactor form one of the central remaining engineering problems around fusion. High temperature fusion plasmas must be surrounded by the walls of a vacuum vessel and confined by strong electromagnetic forces [1]. The heat necessary for fusion reactions presents us with serious requirements on the selection of plasma-facing materials for a thermonuclear fusion reactor. To increase our understanding of plasma-wall interaction, analysis of the wall material is performed. One way of doing this is by using ion beam analysis, more precisely elastic recoil detection analysis (ERDA) [2].

The aim for this project is to analyze and understand the noise of the ion beam analysis technique, ERDA. This can be done by examining two types of models. One of them being classical models such as filters. The other one is our own prediction model, which is a parameterized noise distribution model. Both models will then be assessed for their capability of analyzing the noise.

Prior to this project a lot of research within fusion has been done to construct a reactor that generates energy from nuclear reactions between different hydrogen isotopes. These fusion reactions can produce one million times more energy than chemical reactions. However, the maximum amount of output power current reactors has made has not exceeded 65% of the input power. That is why the International Thermonuclear Experimental Reactor (ITER) is being built in France to create a net energy surplus [4].

The materials analyzed in this project came from the Joint European Torus (JET) reactor in the United Kingdom. The JET, much like the ITER, is a Tokamak type reactor. It is important to note that ERDA is not only used for fusion reactors. It is a method to analysing many kinds of materials, so the advancements made in this project will lead to improvements in analyzing wall materials in general. This could then be applied to fusion-based projects such as the ITER.

II. Theory

In this section the description of what ERDA is, the physics behind the measurements, and how the detectors of the system works are presented.

A. What is ERDA?

Elastic Recoil Detection Analysis (ERDA) is an ion beam analysis technique that measures the elemental composition and depth profiles from thin films and solid bulk samples of arbitrary elements. This technique is also known as forward recoil scattering. In this technique, a high energy ion beam is targeted at a sample and elastic nuclear collision occur between the ions of the beam and the particles of the sample [1].

Under standard conditions you run the ion beam with a range from 10 to 100 MeV. The detector types used for high energy beams include, Time-of-Flight detector, gas ionization chamber and for lower energy beams an implanted silicon detector is typically used [1].

A benefit of ERDA is that you can get information about any element present in the sample from hydrogen all the way to uranium. Another benefit of ERDA is that one can produce simultaneous depth profiles of all elements present in the sample with a single measurement when you use a heavy ion beam and apply time-of-flight detectors. The accessible depth range is typically around 1 μm. ERDA is also very effective in giving information about light elements on heavy substrates [5].

B. The Physics of ERDA

As mentioned in the previous section, ERDA is also known as forward recoil scattering. This means that the measurements are of the recoiled atoms and ions of the sample because of interacting with the projectiles from the ion beam. The projectile ions can be anything from helium ions (alpha particles) up to gold. The recoils come out at an angle larger than 90° with respect to the incoming ion beam trajectory. Even though we measure the particles knocked out of the sample, it is important not to think of ERDA as an analysis based on the sputtering of particles. The depth scale is not obtained by sputtering through the target, but because the projectile ions and the recoils lose energy when penetrating in and out of the sample. In fact, the number of atoms knocked out of the matter is very small and ERDA can be considered a non-destructive technique [1].

There are three main principles of ERDA. The first is that we can get information about the different atomic masses present from the kinematic factor. The second is that we can analyze the atomic composition by using the scattering cross section. The third is that we get a perception of depth based on the stopping power of the sample matter. In ERDA we detect the atoms from the sample, so the signals pertaining to different elements are generated by different recoil atoms. This means that the first set of information about masses can be analyzed without looking at the kinematic factors, instead we can measure the ratio of different masses among the recoil ions [1].

C. Detectors of ERDA

The idea behind ERDA is to measure the recoil energies, but the problems come with identifying which energy signal corresponds to a what element. To differentiate the different recoil species other than looking at their energies, we can use two different methods. The reason we use different methods is since the mass resolution is poorer for heavier recoil species when using the kinematic factor. There is also the problem of overlap of scattered projectiles of recoiled atoms in the energy spectra and that the energy spectra of different recoil atoms may overlap. All this causes a rather complex spectrum [1].

The first way to do this is to simultaneously measure the velocity and the energy of the recoils. Since different masses have a different relationship between the velocity and the energy, this gives a separation of recoils based on their mass. More specifically, these relationships look like banana-shaped regions unique to each mass. Here a time-of-flight detector is used to measure the velocity, therefore this measurement technique is referred to as ToF-E ERDA or just ToF ERDA [6].

We can also use the specific energy loss of a recoil either in a thin solid layer or a gas layer to distinguish between different ion species that exhibits different stopping powers. This gives separation of recoils based on atomic number [5].

For the energy measurement for both these cases we can either use conventional solid-state detector or ionization chamber.

Below is a schematic of the coincidence measurements. Here, there is a beam incident on a sample, and we detect recoils at some fixed angle φ. First the velocity of the recoil is measured by measuring the flight time over some fixed distance L. Then energy is measured in the energy detector. By doing a coincidence measurement where we assume that a flight time signal and energy signal, if they are close enough to each other in time, belong to the same ion, we get for each recoil both the velocity and the energy. This is the type of setup that was used to retrieve the data for this project.

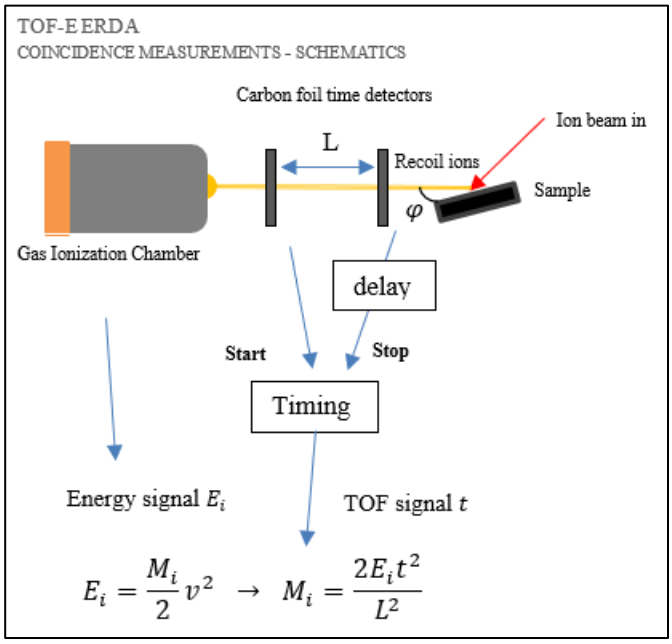


Fig. 1. ToF-ERDA detector setup

III. METHODOLOGY

A. Classical Models

The filters were done using MATLAB. Four filters were produced: A linear filter, a 2D moving average, a combination of both and a median filter. The linear filter removes all the data points with a frequency lower than 1. The 2D-moving average filter takes the average of two adjacent points from the x-axis and two adjacent points from the y-axis while moving through the whole matrix. The median filter goes through the matrix one entry at a time and replaces the new entry with the median of adjacent entries.

B. Prediction Model

To make the parametrized noise distribution, we first had to find where the ideal line for the banana-shaped region would lie. Then make the parametrized noise distribution from the reference samples and fit the measurement from the data set to it. This was done using Python 3. We utilized the software Potku, which is commonly used for ion beam analysis. In Potku the ERDA data is clearly visualized, and you can manually

select which data you want to utilize by marking different points that are connected to each other. A common function in Potku for ERDA is to cut out the banana shaped curve of data points which shows the main relationship between energy and time-of-flight (velocity) of the sample atom being scattered. This is commonly referred to as a cut file.

The simulation of the recoil ions of the data from the reference samples were fitted to the function below.

$$y = \frac{A}{\sqrt{x*B}} + C \ . \qquad (1)$$

Where x represents the raw data for the energy channel, while y is the simulated time-of-flight channel with fitted parameters A, B and C. This was fitted using the cut file of a titanium nitride sample as shown on the two figures below.

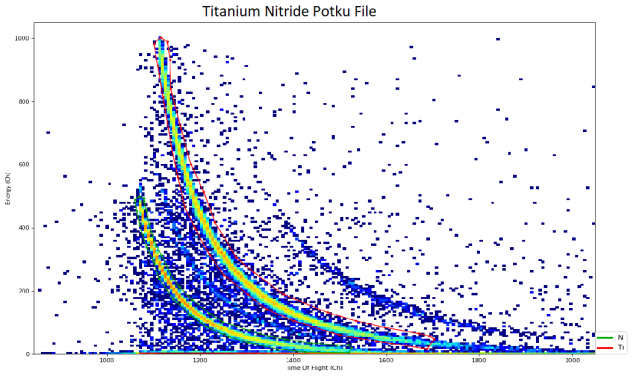


Fig. 2. ERDA of TiN sample using an iodine beam visualized in Potku

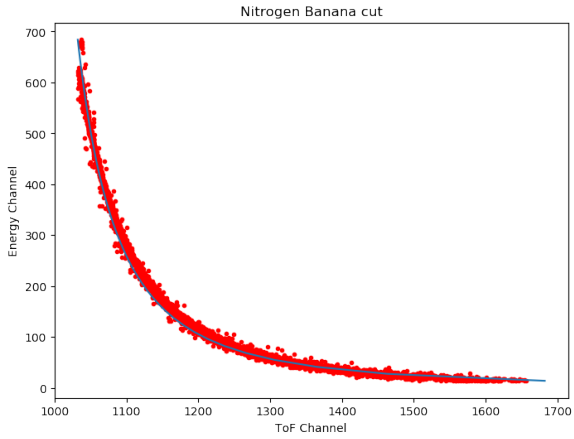


Fig. 3. Simulation of recoil ions of nitrogen using equation 1

The parameterized noise distribution is made by using the function below.

```
def ffunc(x, a, b, c, d, e, f, g):
    y=norm.pdf(x,loc=c,scale=d)*e
    for i in range(len(y)):
        if x[i]<5:
            y[i]+=a
        elif x[i]<c:
            y[i]+=b
        else:
            y[i]+=f*x[i]+g
    return y
```

Fig. 4. Fitting function of the noise distribution

The function has seven parameters (a, b, c, d, e, f and g) and an input variable representing the energy channel data. It returns the yield of each energy level.

Parameter c is the energy level at the center of each banana-shaped region for a given time-of-flight. This number was discovered using the simulation of recoil ion from figure 3.

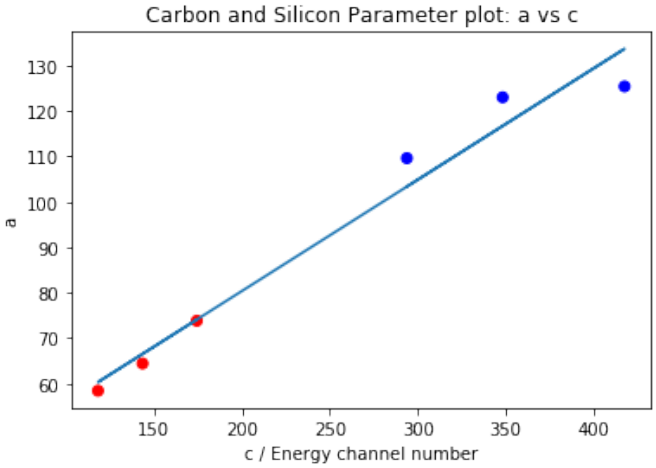


Fig. 5 shows how parameter a in the function varies as parameter c increases for the scattering of both carbon and silicon reference samples. The red points represent the yield for scattered carbon, while the blue points represent the yield for scattered silicon. The left most atom for each element has a time-of-flight channel number of 1200, the middle one has a time-of-flight channel number of 1150 and finally, the right most atom has a time-of-flight channel number of 1120.

Parameter a represents the yield of the energy less than channel number 5. This is meant to represent the particles that go through the time-of-flight detectors but are not detected by the energy detector. Thus, they show up on the ERDA graphs as having an energy of zero, with a variation of time-of-flight. Parameter b represents the yield between energy channel number 5 and the channel number c for each element. Parameter d is the standard deviation of the banana-shaped region since it was assumed to be a normal distribution. Parameter e represents the amplitude of the center of the banana-shaped region. Parameter f represents the slope of the yield in the region with a higher level of energy than the center of the banana-shaped region. Parameter g represents the yield in the region with a higher level of energy than the center of the banana-shaped region.

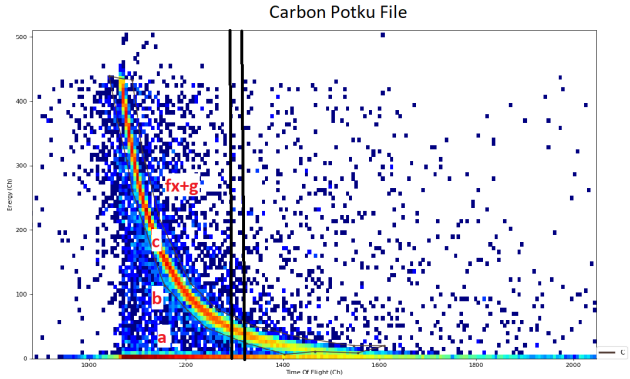


Fig. 6. Visual representation of the location for parameters a, b, c, f and g and a single time-of-flight slice.

Linear trends were shown for the rest of the parameters. Based on these trends we can make a prediction of what the function would look like for a different set of mass and time-of-flight. Therefore, the trends can be used to estimate the noise distribution for several elements. These parameters are applied to titanium nitrate as shown in the results section.

IV. Results

In this section the results of the project are shown. They are divided into two section: the classical models and the prediction model. Section A goes through the classical models while section B presents the prediction model.

A. Classical Models

The following filters are filtering the sample named W109highside, which has tungsten as its main component, but also several other elements.

Figure 6 displays the all the scatter points from the ERDA measurement of W109highside where you can clearly see the banana-shaped trace unique to specific elements combined with a field of noise.

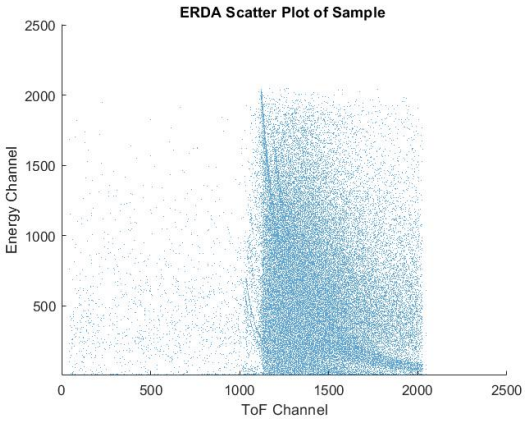


Fig. 6. Scatter plot of sample W109highside

Figure 7 shows the effect of the median filter. Here we see that almost all of the elements were removed except for tungsten and the scattered ions from the ion beam (the main banana-shaped line).

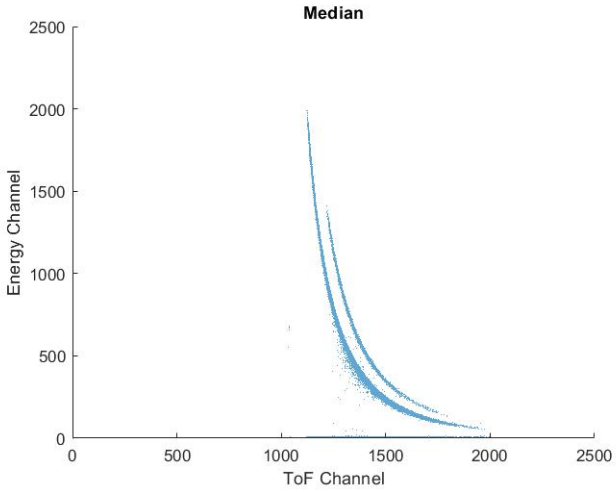


Fig 7. Median filter

In figure 8, the 2-D moving average filter is applied and there is a noticeable trace of one more element to the left of the scattered ion beam shape. There is also some noise left surrounding the center part of the main banana-shape.

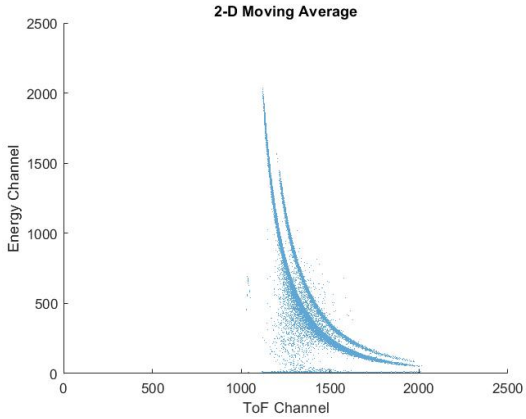


Fig. 8. 2-D moving average filter

For figure 9, the linear filter is applied. It shows a similar pattern to figure 8, but with more noise.

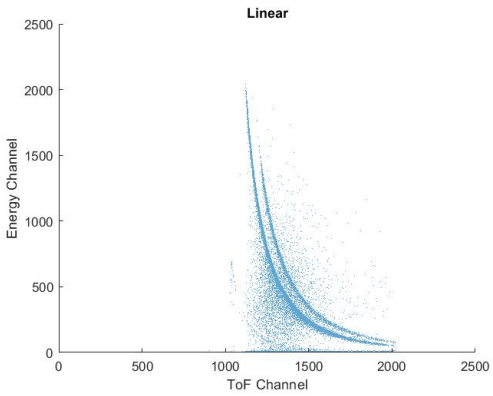


Fig. 9. Linear filter

Finally, for figure 10 it filters out a lot more than the previous two figures with barely any trace of other elemental components.

TABLE I
NOISE DISTRIBUTION PARAMETERS FOR NITROGEN AND TITANTIUM AT A TIME-OF-FLIGHT CHANNEL NUMBER OF 1200

Parameters	Nitrogen	Titanium
a	172.365	58.709
b	0.610	1.797
c	430.987	114.745
d	13.476	5.537
e	1867.694	757.019
f	-0.0001	-9.0496e-05
g	280.761	81.971

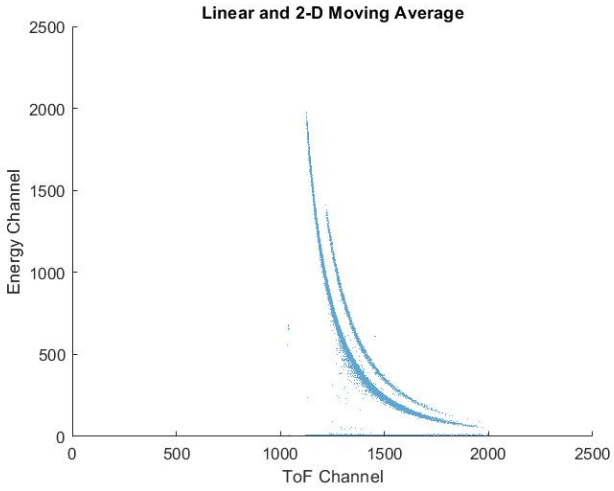


Fig. 10. Linear and 2-D moving average filter.

B. Prediction Model

The following section presents the result of the parameterized noise distribution. Figure 11 shows the result of the parameters being applied to titanium and nitrogen separately. It also includes the actual data from the measured reference sample of titanium nitrate. The parameter results for titanium and nitrogen at a time-of-flight of 1200 (slice from 1195 to 1205) are shown in table 1. Figure 12 then displays the result of the model after a fit to the data has been done.

Similar models were done for a time-of flight of 1120 (slice from 1115 to 1125) and 1150 (slice from 1145 to 1155) for both titanium nitride and aluminum oxide but are not shown in the results section.

Table 2 shows the yield from the banana-shaped cut files for each respective time-of-flight slice. These are thus the yield from the actual reference samples, i.e., the measured number of recoils for each element and time-of-flight. We use the cut files as a comparison to the prediction model to see how much of the noise was used.

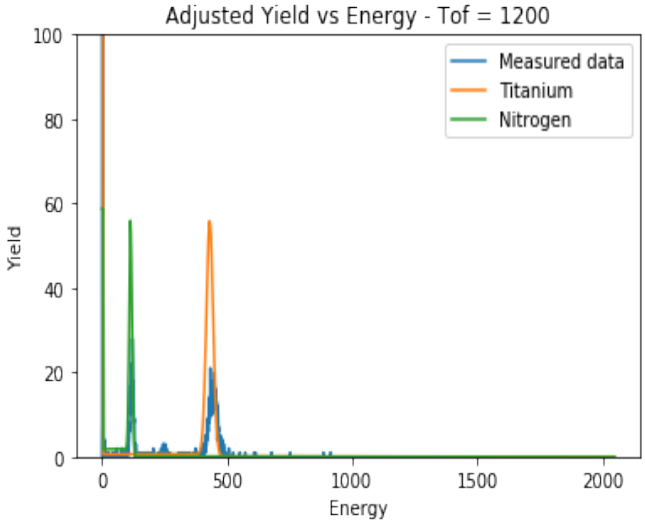


Fig. 11. Parametrized noise distribution of Titanium (orange) and Nitrogen (green) at a time-of-flight channel number of 1200.

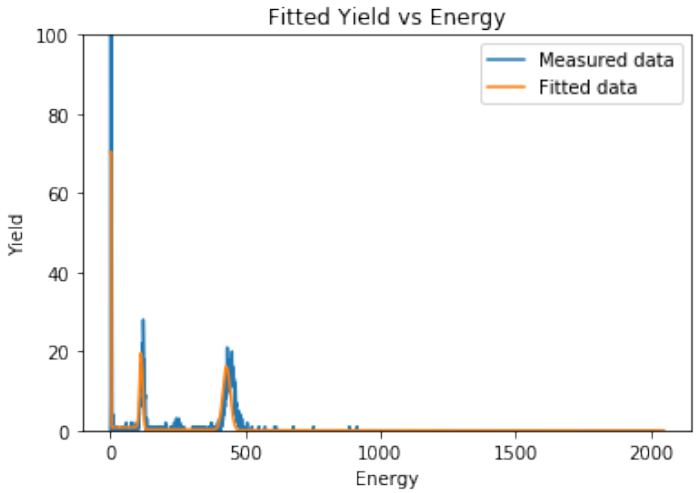


Fig. 12. Parametrized noise distribution of Titanium nitride at a time-of-flight channel number of 1200 after applying fit to real data.

TABLE II
ELEMENTAL YIELD FROM POTKU CUT FILES

ToF / Channel number	Al m = 27 amu	O m = 16 amu	Ti m = 48 amu	N m = 14 amu
1120	169	251	814	701
1150	186	252	824	588
1200	173	222	736	393

Table 3 presents the yield from using the prediction parameters for each element and time-of-flight. Table 4 shows the ratio of the prediction model yield to the cut file yield. Table 5 and table 6 presents the standard deviation for a constant time-of-flight and a constant mass, respectively.

TABLE III
ELEMENTAL YIELD FOR PREDICTION MODEL

ToF / Channel number	Al m = 27 amu	O m = 16 amu	Ti m = 48 amu	N m = 14 amu
1120	201	323	1162	872
1150	226	295	1146	861
1200	247	318	883	455

TABLE IV
PREDICTION MODEL TO CUT FILE RATIO

ToF / Channel number	Al m = 27 amu	O m = 16 amu	Ti m = 48 amu	N m = 14 amu
1120	1.19	1.29	1.43	1.24
1150	1.22	1.17	1.39	1.46
1200	1.43	1.43	1.20	1.16

TABLE V
STANDARD DEVIATION OF YIELD RATIO FOR CONSTANT TIME-OF-FLIGHT

ToF / Channel number	Standard deviation
1120	0.0885
1150	0.1211
1200	0.1261

TABLE VI
STANDARD DEVIATION OF YIELD RATIO FOR CONSTANT MASS

Mass / amu	Standard deviation
N - 14	0.1289
O - 16	0.1068
Al - 27	0.1066
Ti - 48	0.0999

V. Analysis

This section describes and analyzes the results. It will first analyze the classical models, then the prediction model.

For the classical models, four filters were applied: The median filter, 2-D moving average filter, linear filter, and finally a combination of both the linear and 2-D moving average filter.

The yield before and after these filters were applied was never calculated, therefore it is difficult to determine which filter worked best. However, by judging them visually, it looks like the median filter and the combination of the linear and 2-D moving average filters were most effective. After which came the pure 2-D moving average followed by the pure linear filter.

The problem with such filters would be that you exclude a large portion of the noise and data from other smaller specimen, such that the focus lies on the larger banana-shaped regions. One could argue that the banana shape is the only relevant part and therefore these filters are doing effective work. However, the purpose of the project is to analyze and understand the noise, so the filters completely cancel the important information which is the noise. Thus, they are not very useful tools.

Moving on to the prediction model, which is created by the parametrized noise distribution, we see how it fits to the overall data set, including the noise. This model takes the noise and the signal into account when reducing the yield through the fit. By looking at table 2 and 3 we see the varying amounts that the cut files and the prediction models yields for each element. The prediction model has a somewhat higher yield than the cut files. This is because the prediction model utilizes the existing noise and creates a fit, while the cut file is just considering the main banana-shaped region.

When looking at table 5, we can see that the standard deviation increases for increasing time-of-flight. Looking at table 6, we see a trend where the standard deviation decreases as mass increases. This suggests that the precision of the yield increases with a lower time-of-flight (closer to surface of substrate) and a higher mass.

Taking a closer look at table 6, we see that the difference in standard deviation is smaller for aluminum and oxygen, but the difference is greater for nitrogen and oxygen, despite being closer in atomic mass units. This indicates that precision of the yield is more dependent on the sample composition that is being tested than the mass. In this case it was aluminum oxide and titanium nitrate.

VI. Discussion

When taking the classical models into account with the prediction model, we can see that both models have advantages and disadvantages.

The benefits of the classical models are that the figures are visually clearer on where the banana-shaped region occurs, and it considers the whole range of time-of-flight and energy. The drawback is that it completely ignores the noise, which contains important information regarding ERDA and relation to the wall materials. Therefore, for the purpose of this project, these filters are of no interest since they do not provide any information about the noise.

The advantages of the prediction model are that it does take the signal and the noise into account, such that we may use the noise to improve the signals. The limitations are that the range of the time-of-flight is rather small, and this model was only based on two reference samples with three data points each, carbon, and silicon, while also tested on only two samples, aluminum oxide and titanium nitride.

Further investigation with more samples and a greater range of time-of-flight would have been beneficial for this project to have a clearer understanding of the way the noise behaves since it seems to also depend on the composition of each sample that is being tested. Also, the project could have been taken a step further by making an ERDA plot based on the prediction model slices, then comparing the plots before and after applying the model. However, due to the time constraint, such a feat was not possible. This information would be useful for selecting which wall material is to be used in fusion devices. For future projects like this, using machine learning algorithms should be considered for a better mapping of the noise.

VII. Conclusion

In conclusion, both models had advantages and disadvantages, but for the purpose of this project, the filters were rendered useless due to the lack of information regarding the noise, as opposed to the prediction model which takes advantage of the noise. The project needed more time to enhance our models such that we could gain more valuable information regarding the behavior of the noise using ERDA. In the future, a greater number of samples, more analysis and perhaps machine learning algorithms would be beneficial in determining our understanding of wall materials. The increased understanding of wall-materials would benefit the construction of fusion devices such as ITER and JET.

Acknowledgment

I would like to thank my supervisors, Laura Dittrich and Per Petersson for their incredible support throughout this project.

REFERENCES

[1] Strom. P, “Material characterization for magnetically confined fusion: Surface analysis and method development,” Ph.D. dissertation, School. Elect. Eng., KTH., Stockholm., Sweden, 2019.

[2] Mayer, M., Moller, S., Rubel, M., Widdowson, A., Charisopoulos, S., Ahlgren, T., Alves, E., Apostolopoulos, G., Barradas, N.P., Donnelly, S., Fazini´c, S., Heinola, K., Kakuee11, O., Khodja, H., Kimura, A, Lagoyannis, A., Li, M., Markelj, S., Mudrinic, M., Petersson, P., Portnykh,I., Primetzhofer, D., Reichart, P., Ridikas, D., Silva, T., Gonzalez deVicente, S. M., Wang, Y.Q., “Ion beam analysis of fusion plasma-facingmaterials and components: facilities and research challenges,” Nuclear Fusion, vol. 60, pp. 46–48, Dec. 2020

[3] Malmqvist, M. (2019, Jun.) Fusionplasma device extrap t2r. KTH, Stockholm, Sweden. Accessed: May 2, 2021. [Online]. Available:

[4] Hong, Q., Davidson, R., Startsev, E., Lee, W., “f simulation studies of the ion–electron two-stream instability in heavy ion fusion beams,” Laserand Particle Beams, vol. 21, pp. 21–26, Jul. 2003

[5] Bieniosek Ernest, F.M.,Beam imaging diagnostics for heavy ion beamfusion experiments. Portland, USA: Orlando Lawrence Berkeley NationalLaboratory, May 2003

[6] Rubel, M., Petersson, P., Alves, E., Brezinsek, S., Coad, J., Heinola, K., Mayer, M., Widdowson, Anna., “The role and application of ion beam analysis for studies of plasma-facing components in controlled fusion devices,” Nucl. Instrum. Methods Phys. Res. B: Beam Interactions with Materials and Atoms, vol 371, pp. 4-11, Mar. 2016

[7] Primetzhofer, D., Strom, P., Petersson, P., “Ion Beam Materials Analysis,” .,Uppsala University., Uppsala., Sweden, 2019.

Modeling of RF Heating in the JET Tokamak

Wilhelm Holmberg and Emil Söderman

***Abstract*—Fusion reactors need methods to couple external power to confined plasmas. Ion cyclotron resonance heating (ICRH) is a method to radiate electromagnetic waves to couple power to the kinetic motion of gyrating ions in a plasma. In this report we study ICRH with regard to the dispersive effects of the confined plasma in the JET tokamak. We study this with a division of the electric field with regard to toroidal mode numbers. Specifically we examine where dispersive effects are located in the plasma and if they have any importance. We also study the ion-ion hybrid layer, where according to the dispersion relation a singularity can occur. To do the analysis, we use the code FEMIC to simulate fusion scenarios.**

The results show that ion absorption is stronger and more localized for low toroidal mode numbers. This is true for toroidal mode numbers $n_\phi \geq 10$, and for $n_\phi < 6$ the resolution of the solutions does not suffice for a meaningful analysis. An examination of the effects of the singularity in the dispersion relation at the ion-ion hybrid layer shows that the amount of absorbed power in the ion-ion hybrid layer is significantly smaller than for the central region of absorption. This means that the singularity does not affect ICRH heating in our scenarios in any major sense.

***Sammanfattning*—Fusionsreaktorer behöver metoder för att överföra extern effekt till ett inneslutet plasma. Joncyclotron-resonansuppvärmning (ICRH) är en uppvärmningsmetod där man strålar elektromagnetiska vågor in i ett plasma, vars effekt överförs till roterande joner. I denna rapport studerar vi ICRH med avseende på de dispersiva effekterna i ett inneslutet plasma i tokamaken JET. Detta undersöks med en uppdelning av det totala elektriska fältet med avseende på toroidala moder. Vi undersöker specifikt var i plasmat de dispersiva effekterna uppstår och analyserar deras påverkan på effektöverföringen. Vi studerar även en singularitet som uppstår i dispersionsrelationen vid jon-jon-hybridlagret. För att göra en analys används koden FEMIC för att simulera ICRH-scenarion.**

Resultaten visar att jonernas absorption är starkare och mer lokaliserad för låga toroidala modtal. Detta är sant för modtal $n_\phi \geq 10$. För $n_\phi < 6$ är lösningarnas upplösning för dålig för en betydelsefull analys. En undersökning av jon-jon-hybridlagret visar att mängden absorberad effekt inom detta område är avsevärt mindre än för de centrala områdena för absorption. Detta medför att singulariteten i dispersionsrelationen inte påverkar ICRH-uppvärmningen i våra simuleringar till en betydelsefull grad.

***Index Terms*—fusion, tokamak, ICRH, RF heating, JET**

Supervisors: Thomas Jonsson and Björn Ljungberg

TRITA number: TRITA-EECS-EX-2021:162

I. Introduction

A. Energy

Society runs on energy. Today, a majority of that energy is generated by fossil fuels, see Fig. 1. Burning fossil fuels leads to large CO_2 emissions, which in turn leads to climate change. To combat this, a great effort is being done to replace fossil fuels with renewable electricity production, electrifying transportation and even replacing coal with hydrogen in steel production [2]. This does however, require a lot more and reliable electricity production. Wind and solar are both renewable energy sources, but they are intermittent and rely on the weather, which means that the energy either need to be stored or backed up by another energy source that can fill in the gap of their production. Today there are fission nuclear plants that have stable energy production, but they also produce long lived radioactive waste.

B. Fusion

Fusion energy comes from the same process that stars use to generate energy. During the process, elements fuse and form heavier elements. For lighter elements, the end products have less total mass than they began with [3]. This difference in mass equals the energy released by the process and can be described by

$$\Delta E = \Delta m c^2, \tag{1}$$

where ΔE is the energy released, Δm is the difference in mass and c is the speed of light in vacuum.

One of the easiest and most energy efficient [4] fusion reactions is that between the hydrogen isotopes deuterium (^{2}H or D) and tritium (^{3}H or T).

$$\mathrm{D} + \mathrm{T} \rightarrow {}^4\mathrm{He} + \mathrm{n} + 17.6\,\mathrm{MeV}. \tag{2}$$

As can be seen in (2), the end product of this reaction produces one helium atom, one neutron and releases $17.6\,\mathrm{MeV}$ of energy for each D and T that fuses. The free neutron can be a problem here as it can activate the surrounding material making it radioactive [3]. To solve this, there are attempts to

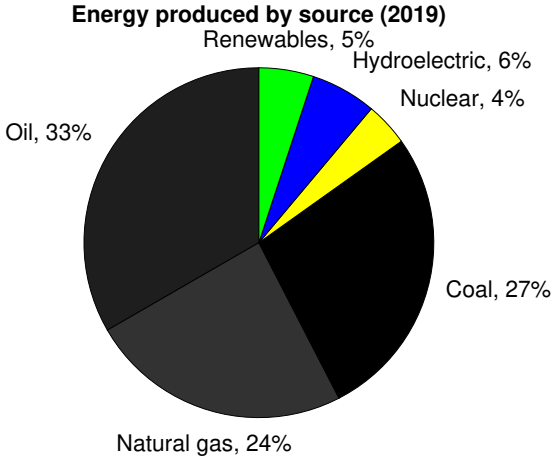


Fig. 1: Energy produced by source [1].

line the reactor with a blanket of lithium that, when hit by a neutron, decays into helium and tritium. This also produces more fuel that can be fed back into the reactor [3]. The blanket won't capture all the neutrons and there will still be a small amount of radioactive material that need to be handled. The radioactive materials are however relatively short lived with a life-time of around 100 years compared to that of around 100 000 years from for example a fission reaction [3]. For fusion to occur the Coulomb force needs to be overcome. For this to happen, the hydrogen isotopes need to be heated to at least 100 million K [4]. As the hydrogen heats up, it ionizes and becomes a plasma.

C. Magnetic Confinement

No material can withstand temperatures at 100 million K, therefore the plasma needs to be confined in some way. One way of doing this is with magnetic confinement, where the hot plasma is confined within a vacuum vessel. This means that the plasma is held together by strong magnetic fields (of the order of several teslas) [4].

D. Tokamaks

The most well researched [4] type of fusion reactor is the tokamak. The tokamak is a torus or a doughnut shaped reactor that uses a magnetic field coils to create a helical magnetic field containing the plasma, as illustrated in Fig. 2.

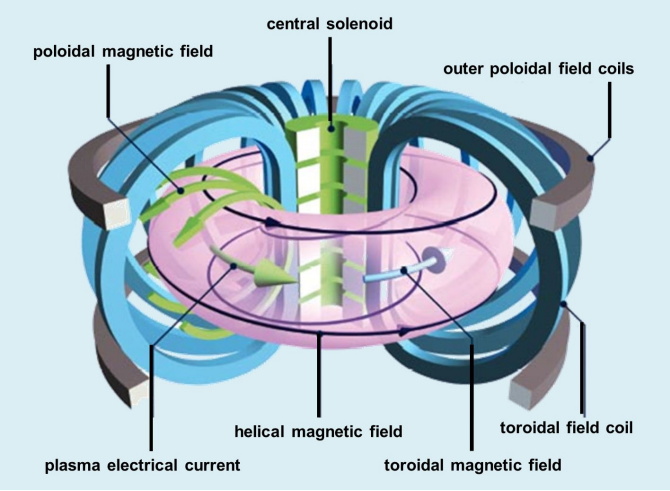


Fig. 2: Illustration of a tokamak reactor [5].

The magnetic field created by the field coils contributes to a Lorentz force, see (3), that makes the charged particles move in a gyro motion around the magnetic field lines.

$$\mathbf{F} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B}). \tag{3}$$

The frequency of which the particles rotate around the magnetic field lines is called the cyclotron frequency. The force from the electric field is in this case small compared to the force from magnetic field, leading to the magnetic force being dominant.

The largest operational tokamak is the Joint European Torus (JET) located in the UK and will be the reactor that this study focuses on.

One of the big problems that fusion power faces is that it has never produced more power then used to heat the plasma. JET set the current record when it in 1997 produced $16\,\mathrm{MW}$ of thermal power from fusion with $24\,\mathrm{MW}$ heating power, resulting in a gain factor Q = 0.67 [6]. ITER ('The way' in latin), which is JET's successor is currently being built in the south of France. Scientists aim to achieve a Q = 10 with the ITER reactor, meaning it will produce 10 times more power than used to heat it [4].

E. Heating Plasma

There are several different methods for heating plasmas. The three most common methods used are ohmic heating, neutral beam injection (NBI) and radio frequency (RF) heating [3]. Ohmic heating is possible due to the resistivity of the plasma, but as the temperature becomes higher, this method becomes less effective since the resistivity R of the plasma decreases with the temperature T as $R \propto T^{-3/2}$ [4]. NBI uses fast neutral particles (typically hydrogen) that are injected into the plasma at high velocity [3]. Once injected the particles become ionized due to collisions and can be confined by the magnetic field [4]. The fast ions can then transfer their kinetic energy through Coulomb collisions with the plasma ions.

In RF heating an antenna sends electromagnetic waves into the plasma where the wave energy is absorbed. RF heating affects particles differently depending of which frequency is used, which means that several subcategories of RF heating exist [3]. One example of this is ion cyclotron resonance heating, ICRH [4]. The ions of the plasma gyrate around the magnetic field lines around the tokamak with a cyclotron frequency. If the antenna frequency matches the cyclotron frequency, the electromagnetic wave will interact with the plasma and accelerate the ions increasing their kinetic energy, which heats the plasma [4].

F. Goals

This study will be simulating ICRH heating in the JET tokamak. Firstly, parameters will be chosen to describe, qualitatively and quantitatively, how the electromagnetic waves travel in the plasma and how they are absorbed. This study aims to answer the following questions:

- Where does dispersive effects appear in the plasma and how important are they?
- How do different toroidal mode numbers affect the heating ability of the waves and how much does the heating vary?
- How are the waves affected by a potential singularity in the ion-ion hybrid layer?

Lastly we aim to calculate a 3D electric field by summing a series of 2D electric field components and analyze the significance of these components.

II. Theory

A. Coordinates

One usually uses cartesian, cylindrical, or toroidal coordinates to describe position in space in a tokamak. In Fig. 3

we see both toroidal and cylindrical coordinates. The cylindrical coordinates are $[R, \phi, Z]$ and the toroidal coordinates $[r, \phi, \theta]$. The relations between the two coordinate systems are $R = R_0 + r\cos(\theta)$ and $Z = r\sin(\theta)$, where R_0 is the distance from $R = 0$ to the center of the tokamak cross section. In this report we will call the ϕ-direction, the toroidal direction and the θ-direction, the poloidal direction.

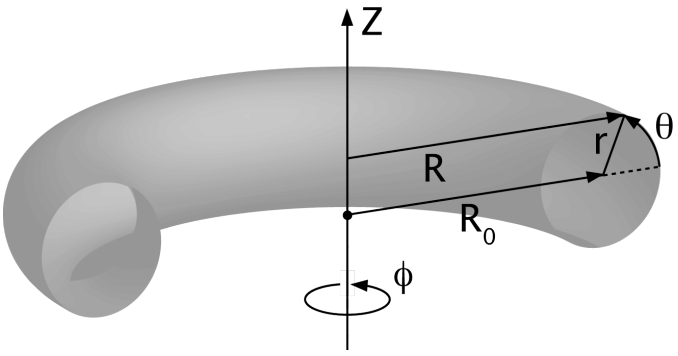


Fig. 3: Toroidal coordinate system [7].

B. Waves in Plasma

We model the propagation of the electromagnetic waves in a plasma through the wave equation in frequency domain and make the ansatz that we will get a solution

$$\mathbf{E}(\mathbf{r}, \omega) = \mathbf{E}_0(\mathbf{r})e^{-i\omega t}. \tag{4}$$

We can thus write the wave equation

$$\nabla \times (\nabla \times \mathbf{E}) - \frac{\omega^2}{c^2}\mathbf{E} = i\omega\mu_0(\mathbf{J}_{\text{ind}} + \mathbf{J}_{\text{ext}}), \tag{5}$$

where $\mathbf{J}_{\text{ind}}$ is the induced plasma current density and $\mathbf{J}_{\text{ext}}$ is the current in the antenna. The induced plasma current density $\mathbf{J}_{\text{ind}}$ is related to the electric field by Ohm's law $\mathbf{J}_{\text{ind}} = \tilde{\boldsymbol{\sigma}}\mathbf{E}$, where $\sim$ symbolizes operator. If we substitute this into the wave equation we get

$$\nabla \times (\nabla \times \mathbf{E}) - \frac{\omega^2}{c^2}\tilde{\mathbf{K}}\mathbf{E} = i\omega\mu_0\mathbf{J}_{\text{ext}}, \tag{6}$$

where the $\tilde{\mathbf{K}}$ tensor is

$$\tilde{\mathbf{K}} = \mathbf{I} + \frac{j}{\varepsilon_0\omega}\tilde{\boldsymbol{\sigma}} \equiv \mathbf{I} + \tilde{\boldsymbol{\chi}}, \tag{7}$$

and $\tilde{\boldsymbol{\chi}}$ is the susceptibility tensor [4]. We call $\tilde{\mathbf{K}}$ the dielectric tensor and it describes the electromagnetic properties of a plasma.

C. Modeling of plasma

In fusion plasma physics a simplified cold plasma model is sometimes used. The cold plasma model assumes that the plasma particles are stationary. If in contrast to cold plasma, the velocities of the particles are accounted for, the warm plasma model is obtained.

The electromagnetic response of each particle can be described as the cold plasma response with a Doppler shift. This means that the dielectric tensor will be

$$\mathbf{K}(\omega, \mathbf{k}) = \sum_j \mathbf{K}_{\text{cold}}(\omega - \mathbf{k}\cdot\mathbf{v}_j), \tag{8}$$

where $\mathbf{v}_j$ is the velocity of a particle and j denotes a summation over every particle in the plasma [8]. The warm plasma tensor is a rank 2 tensor, which means it will have 9 entries

$$\mathbf{K} = \begin{bmatrix} K_{xx} & K_{xy} & K_{xz} \\ K_{yx} & K_{yy} & K_{yz} \\ K_{zx} & K_{zy} & K_{zz} \end{bmatrix}. \tag{9}$$

In this project we model a warm magnetized plasma. The task of tracking each particle in a warm magnetized plasma is impossible, which is why we rather use statistical methods to derive the dielectric tensor. The warm magnetized plasma tensor depends on the properties of the ion species, frequency and spatial coordinates. The explicit expression of the tensor is rather hefty but can be examined in Swanson's book on plasma waves [9].

D. Fourier Decomposition

For a toroidal plasma in cylindrical coordinates we can assume that the plasma response and geometry are symmetric with regards to the toroidal axis. If we Fourier decompose the electric field in the toroidal direction it yields a solution as in (10).

$$\mathbf{E}_{\text{tot}}(R, \phi, Z) = \sum_{n_\phi=-\infty}^{\infty} \hat{\mathbf{E}}_{n_\phi}(R, Z)e^{in_\phi\phi}. \tag{10}$$

This means that we can examine electric field components $\hat{\mathbf{E}}_{n_\phi}$ with regard to toroidal mode numbers n_ϕ.

E. Dispersion Relation

A dispersion relation describes the relation between a wave number k, and angular frequency ω. The wave number k describes the properties of an electromagnetic wave with respect to propagation and decay in the plasma. The wave number is complex, where $\text{Re}\{k\}$ describes the propagation of the wave and $\text{Im}\{k\}$ the decay of the wave.

Due to the geometry of the tokamak, we divide the the wave vector into a parallel and a perpendicular component with respect to the toroidal direction. We can approximate the wave numbers as

$$k_\parallel = \frac{n_\phi}{R}, \tag{11}$$

for the parallel wave number and

$$k^2_{\perp,\text{FW}} \approx \frac{\omega^2}{c^2}\left[K_{yy} - n^2_\parallel + \frac{K^2_{xy}}{K_{xx} - n^2_\parallel}\right], \tag{12}$$

for the perpendicular wave number. Here $n_\parallel = ck_\parallel/\omega$ represents the plasmas refractive index component in the toroidal direction. Note that the wave numbers are dependant on n_ϕ. Equation (12) is the dispersion relation used in our analysis.

In equation (12) we can see that if the denominator $K_{xx} - n_{\parallel}^2 \to 0$ in the third term approaches zero, the wave number approaches a singularity which has significance in the analysis of this report. The region where this occurs is called the ion-ion resonance layer. K_{xx} is complex, which means that for a true singularity to occur, the imaginary part also has to approach zero. This only happens if $n_\phi = 0$. For small n_ϕ, the wavelengths of the EM waves becomes very short, which means that $k_\perp$ becomes very large.

F. Ion Cyclotron Resonance Heating

The purpose of ICRH is to transfer energy from electromagnetic waves to the kinetic energy of ions and electrons. In a tokamak plasma the ions as well as the electrons are subject to both a poloidal and a toroidal magnetic field, which sum up to a helical magnetic field. Electromagnetic field theory tells us that charged particles are subject to the Lorentz force

$$\mathbf{F} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B}) = m\frac{d\mathbf{v}}{dt}. \tag{13}$$

If in a tokamak the effect of the electric field on the plasma particles is significantly smaller than the effect of the magnetic field, one can discard the electric field in calculations. The particles thus gyrate around the magnetic field lines in the tokamak with an angular frequency called the cyclotron angular frequency and is given by

$$\Omega = \frac{q|\mathbf{B}|}{m}. \tag{14}$$

This gives rise to a resonance phenomenon, called cyclotron resonance, with electromagnetic waves, which occurs when

$$\omega = n\Omega + k_\parallel v_\parallel, \tag{15}$$

where n is the harmonic number and $v_\parallel$ is the parallel velocity of the ion.

G. Wave Propagation and the Dispersion Relation

A tokamak plasma is magnetically confined in a vacuum toroidal chamber. The plasma particles are not homogeneously distributed in the tokamak, which gives regions of different plasma densities and temperatures throughout the tokamak. This means that the electromagnetic waves will have different responses throughout the plasma.

In ICRH the objective is to radiate electromagnetic waves from an antenna attached to the tokamak wall and let the waves propagate into the center of the plasma where they will resonate and be absorbed.

The poloidal cross section of the tokamak can be plotted with magnetic flux surfaces as in Fig. 4. The outermost flux surface is called the separatrix. The area between the tokamak walls and the separatrix is called the scrape-off layer, SOL. This layer is characterized by very low plasma densities and can practically be modelled as a vacuum. If the wavelengths of the emitted waves are long, the wave number in the direction pointing towards the plasma from the antenna will become purely imaginary in the SOL and the waves will tunnel through the SOL. This can be explained by

$$k^2 = k_\phi^2 + k_x^2, \tag{16}$$

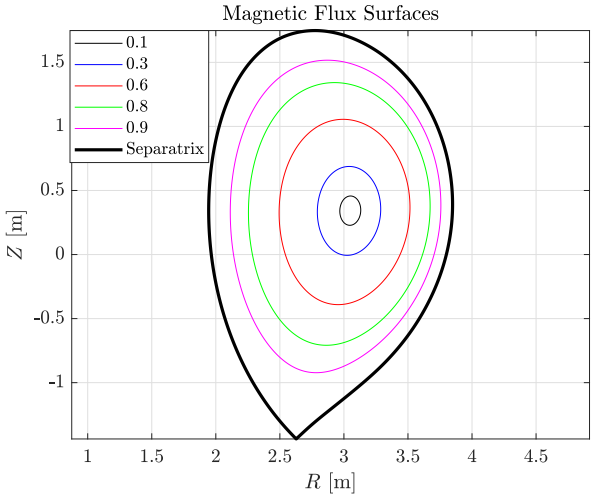


Fig. 4: Magnetic flux surfaces plotted on the poloidal cross section of the tokamak. The values in the legend correspond to a normalized radius ρ_{pol}, $\rho_{\text{pol}} \in [\rho_{\text{axis}}, \rho_{\text{separatrix}}] = [0, 1]$.

where $k = \omega/c$, c is the speed of light in vacuum, $k_\phi = n_\phi/R$ is the toroidal wave number and k_x is the component in the direction pointing towards the plasma, which leads to

$$k_x = \sqrt{k^2 - k_\phi^2}. \tag{17}$$

This means that for large n_ϕ, k_x will be imaginary and the waves will tunnel through the SOL. The consequence of this is that the wave is not propagating, but is decaying exponentially. It does however begin to propagate inside of the separatrix where the plasma density is higher, but with lower amplitude due to the exponential decay in the SOL. The propagation of the waves inside the plasma depends on the dielectric tensor and the magnetic field.

III. Method

A. FEMIC

In this report our aim is to account for dispersive effects in ICRH heating in the JET tokamak. To do this we have used the MATLAB [10] based code FEMIC (Finite Element Method for Ion Cyclotron heating) [11] to generate fusion plasma scenarios in the finite element method based simulation program COMSOL Multiphysics [12]. The FEMIC package includes a CAD model of the JET reactor including models of the ICRH antennas. In our case we have used the JET ITER like antenna (JET ILA).

B. Scenario

We have chosen to simulate a scenario with a plasma constituting of 97% deuterium with a minority of 3% hydrogen. The density and temperature profiles used in the simulations at $Z = 0$, can be seen in Fig. 5 and 6.

The frequency of the antenna was set to center resonances and absorption. This was done by calculating the resonance frequency of hydrogen using equation (14) and setting $\Omega = \omega_c$, where the toroidal magnetic field at the center $R = R_0 = 3.05\,\text{m}$ is given by FEMIC as $|\mathbf{B_0}| = 2.7\,\text{T}$, q is the

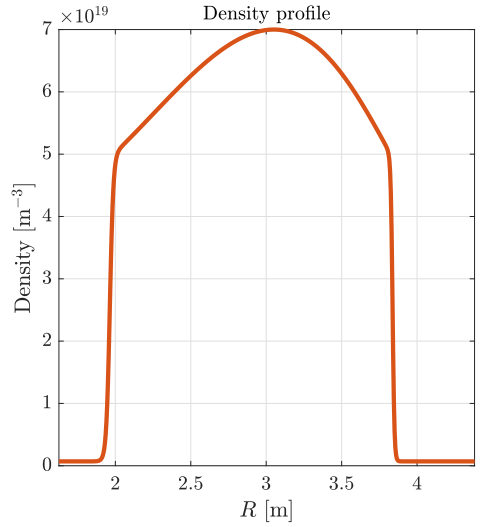


Fig. 5: Density profile at $Z = 0$, as used in simulations.

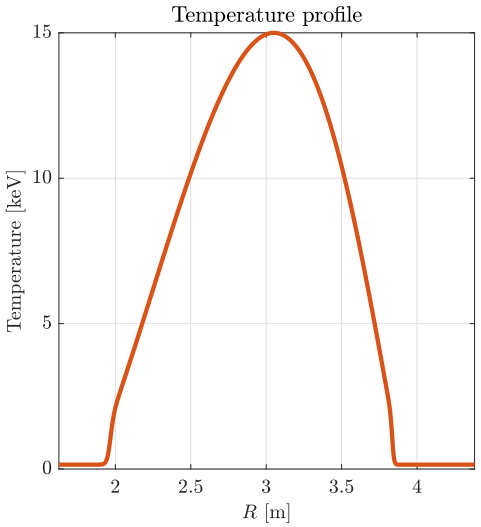


Fig. 6: Temperature profile at $Z = 0$, as used in simulations.

elementary charge and m is the mass of hydrogen. This lead to $f_c = 41.151\,\mathrm{MHz}$.

In this study, the main parameter of interest is the toroidal mode number n_ϕ. In every simulation an antenna unit surface current density of $1\,\mathrm{A/m}$ has been used.

The parameters chosen to analyse, qualitatively and quantitatively, how the electromagnetic waves travel and how they are absorbed in the plasma were:

- Total power absorption.
- Species absorption.
- The width and centering of absorption.
- The imaginary part of the dispersion relation.

The goal of ICRH is to transfer as much power as possible to the middle of the plasma. One wants to heat the middle of the plasma due to the heat diffusing slower to the separatrix from the center of the plasma than for the outer regions of the plasma. The absorption of the different species also becomes important due to the electrons absorbing power all throughout the plasma instead of in the resonance regions.

IV. Results

The simulation results are presented here. First we present the dispersive effects with regard to wave propagation by plotting the dispersion relation. To do a quantitative analysis, three $k_\perp$ plots were added together, which makes it possible to see how the wave properties change with respect to n_ϕ. The electric field strength was also plotted for different toroidal mode numbers. With these plots we can do a qualitative analysis on how the electric field changes for different n_ϕ.

In the second part, to see how the wave power absorption is affected by the parameters, we do a qualitative analysis of power absorption plots to see where the power is absorbed. These plots show the centering of absorption.

For quantitative analysis we plot the total power absorption, to see how much is absorbed per toroidal mode number. We also plot the partition of absorption with regard to the different plasma particles, i.e hydrogen, deuterium and electrons. This is done for $n_\phi \in [6, 50]$. We also study how much power is absorbed within a certain radius to quantify absorption width.

To study the ion-ion hybrid layer, where $K_{xx} - n_\parallel \to 0$, we first had to set a finer resolution to the mesh grid to see the effects better. We chose to study $n_\phi = 8$, because effects are more prominent for lower toroidal mode numbers, but for $n_\phi < 8$, we get problems with the resolution of the solution.

Lastly, to create a 3D left hand polarized E-field, a MATLAB script ran for all $n_\phi \in [-50, -6] \cup [6, 50]$, to save all the data from the simulated 2D E-fields. We excluded $n_\phi \in [-5, 5]$, due to the resolution problems. The resolution problems is due to the ion-ion hybrid layer resonance approaching a singularity for short wavelengths. The antenna spectrum for the JET ILA was then extracted and the antenna current for each toroidal mode number j_{n_ϕ} was then multiplied with the electric field component for each toroidal mode number. In Fig. 7 the current spectrum can be seen. This was possible because every simulation ran with unit surface current density in the antenna. The resulting matrix was then used to plot slices of the electric field in the reactor as a Fourier sum over toroidal mode numbers as in equation (10).

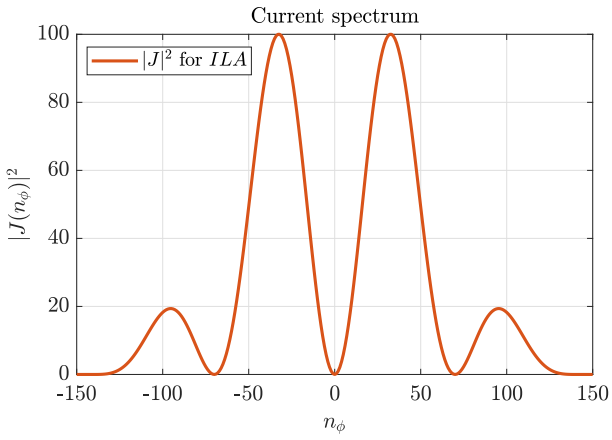


Fig. 7: The antenna current spectrum for the JET ILA antenna.

A. Wave Propagation

The propagation of the electromagnetic waves in the JET tokamak is shown in Fig. 8, represented by the perpendicular wave number where $\mathrm{Re}\{k_\perp\}$ is the propagation and $\mathrm{Im}\{k_\perp\}$ the decay. The plot shows the propagation and decay for three toroidal mode numbers $n_\phi = 10$, 32 and 50.

In Fig 8, we see that the $\mathrm{Re}\{k_\perp\} = 0$ and $\mathrm{Im}\{k_\perp\} \neq 0$ for toroidal mode numbers larger than 10, in the grey right part of the graph ($R > 3.75\,\mathrm{m}$). This means that the waves tunnel through the separatrix for these modes as expected by theory. When the waves tunnel, their amplitude decreases exponentially and that is important as it determines how much power enters the plasma.

In figures 9, 10 and 11, electric fields are plotted for the JET tokamak's poloidal cross section in contour plots. The plot in Fig. 9, demonstrates that the decay of the EM waves are larger in the central region of the plasma for $n_\phi = 10$, than for the two other toroidal mode numbers shown in Fig. 10 and 11.

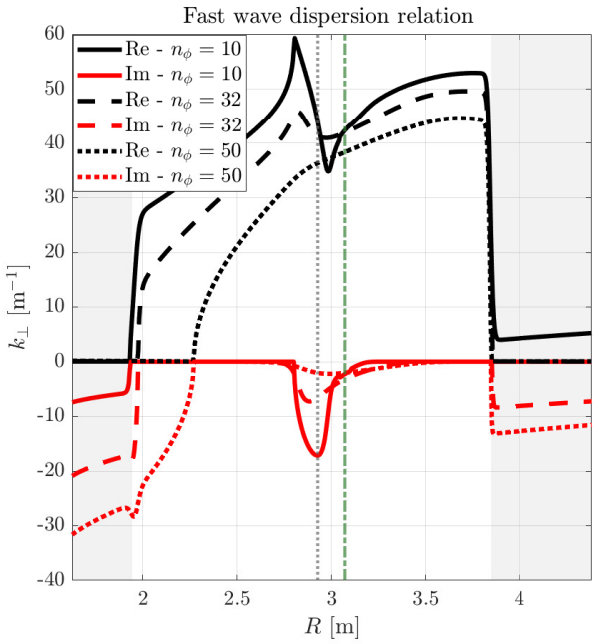


Fig. 8: Dispersion relation for $n_\phi = 10$, 32 and 50. The perpendicular wave number $k_\perp$ is plotted as a function of the major radius in the equatorial plane. The grey line represents where the ion-ion hybrid layer occurs for $n_\phi = 10$. The green line represents where ion resonances are located for $n_\phi = 10$, where in this case approximately coincide for hydrogen and deuterium.

For all contour plots we see standing wave patterns in the plasma, seen as oscillations in the wave strength. The plot for $n_\phi = 10$ in Fig. 9, shows that the electric field is concentrated to the right part of the plot, close to the antenna. One can also notice a faint line crossing a little left of the axis, which is where $K_{xx} = n_\parallel$. This line is the ion-ion hybrid layer. There is also a ripple effect in Fig. 9, which is a standing wave that is caused due to it being reflected at the ion-ion hybrid layer. For higher mode numbers in Fig. 10 and 11, we see that the electromagnetic waves have propagated to the left of the center. For Fig. 10 the wave makes it all the way through the plasma before being reflected at the separatrix. In Fig 11, the waves are reflected inside the plasma, at around $R \approx 2.3\,\mathrm{m}$. This can also be seen in Fig. 8 as the $\mathrm{Re}\{k_\perp\} = 0$ and $\mathrm{Im}\{k_\perp\} \neq 0$ for $n_\phi = 50$, which means that the wave does not propagate for $R < 2.3\,\mathrm{m}$.

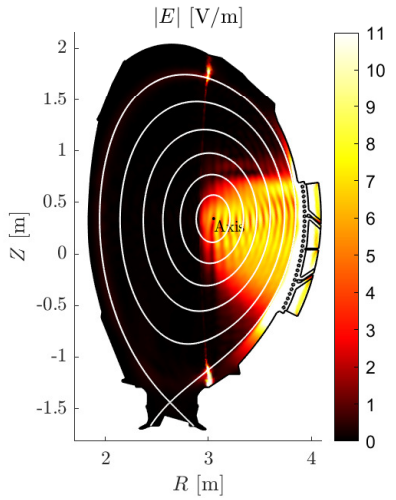


Fig. 9: Electric field strength magnitude plotted for $n_\phi = 10$.

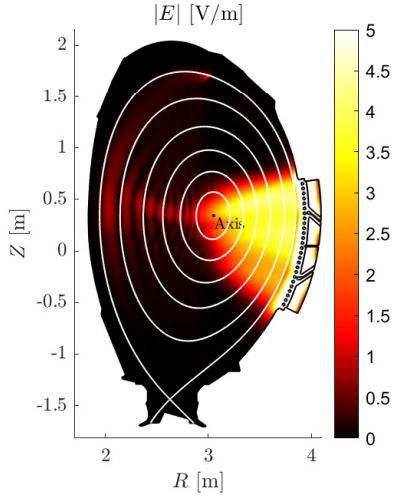


Fig. 10: Electric field strength magnitude plotted for $n_\phi = 32$.

B. Absorption

The power absorption of the electromagnetic waves with regard to toroidal mode number can be seen in figures 12, 13 and 14, for $n_\phi = 10$, 32 and 50 respectively. It can also be seen that the width of the absorption varies. The total power absorption decreases with higher mode numbers.

It can be seen that $n_\phi = 10$ gives the highest total power absorption. The power absorption for $n_\phi = 10$ had narrow absorption bands centered around the resonance areas of hydrogen and deuterium, see Fig. 12.

In Fig. 13 it can be seen that the absorption is more spread out but mostly still in the middle of the plasma. The total amount of absorbed power is less and it can be seen in the amplitude scale, which has lower peaks. This is due to the

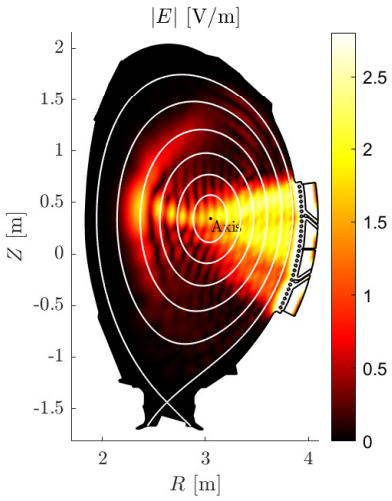


Fig. 11: Electric field strength magnitude plotted for $n_\phi = 50$.

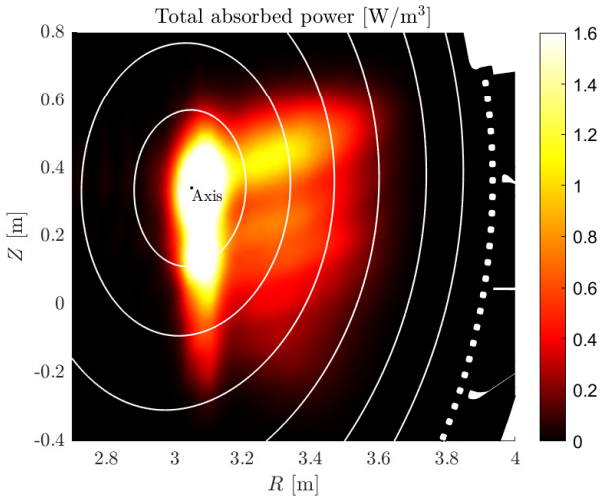


Fig. 13: Power absorption for $n_\phi = 32$.

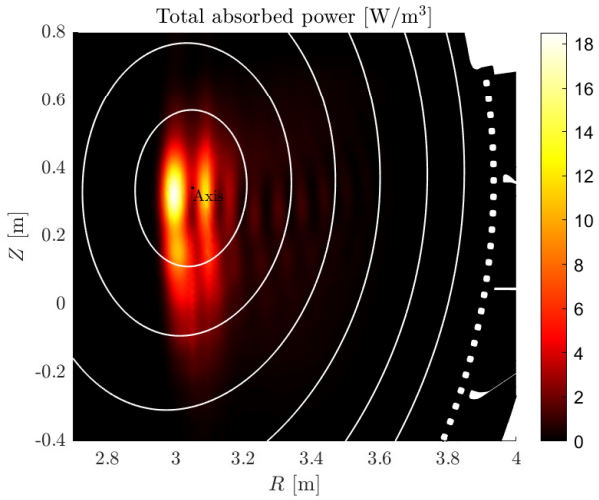


Fig. 12: Power absorption for $n_\phi = 10$.

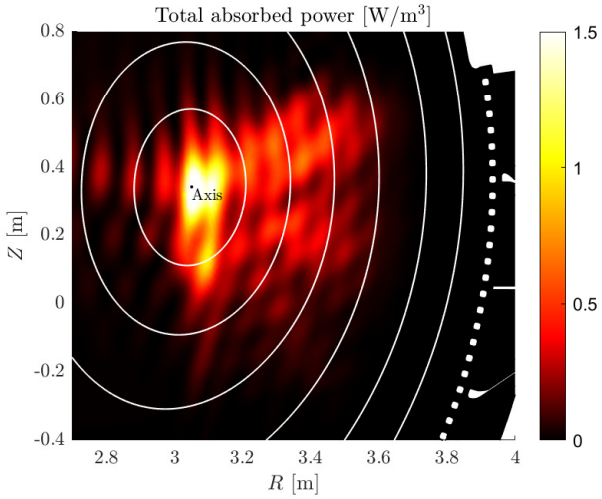


Fig. 14: Power absorption for $n_\phi = 50$.

tunneling in the SOL. For $n_\phi = 32$ the absorption is wider and has no clear boundaries like $n_\phi = 10$ does.

The absorption for $n_\phi = 50$ is very widespread in relation to $n_\phi = 10$. The total amount absorbed is also substantially lower, again this is due to the tunneling through the SOL.

The width of absorption of all 3 modes can be seen in Fig. 15, where total absorbed power is shown as a function of the radius from the center to the separatrix.

In Fig. 16 the total power absorption can be seen as a function of n_ϕ, when n_ϕ runs from 6 to 50. One can notice that the power absorption peaks at $n_\phi = 9$ and 10, and decreases as the toroidal number increases.

In Fig. 17 the partition of which percentage of power the hydrogen, deuterium and electrons absorb of the electromagnetic waves can be seen. For low n_ϕ the hydrogen absorbs most of the power. At $n_\phi = 18$, we see a maximum in the electron absorption, absorbing roughly 50% of the total power. For $n_\phi \gtrsim 44$, the three species absorb roughly an equal amount of power.

C. Ion-ion Hybrid Layer

The ion-ion hybrid layer can be seen in Fig. 18 and 19 as a distinct line left of the axis. In Fig. 20, one can see where the absorption occurs in the reactor. This is quantified in Fig. 21, which is zoomed in and saturated in the colour scale. One can conclude that no meaningful amount of power gets absorbed in the ion-ion hybrid layer. Most power is absorbed in the resonance area. This means that the ion-ion hybrid layer does not have any major impact for $n_\phi = 8$, thus mathematical errors caused by it are minor.

Reflections of the waves in the ion-ion hybrid layer for $n_\phi = 8$ create a standing wave pattern, see Fig. 18, similarly as for $n_\phi = 10$, observed in Fig. 9. The standing waves are even more prominent in $n_\phi = 8$ than for $n_\phi = 10$. We can also conclude that the ion-ion hybrid layer is most prominent for low toroidal mode numbers and vanishes for higher modes.

In Fig. 22 we can see the resolution problems for the lower n_ϕ, we see that the plot become too grainy to properly analyse it in the ion-ion hybrid layer.

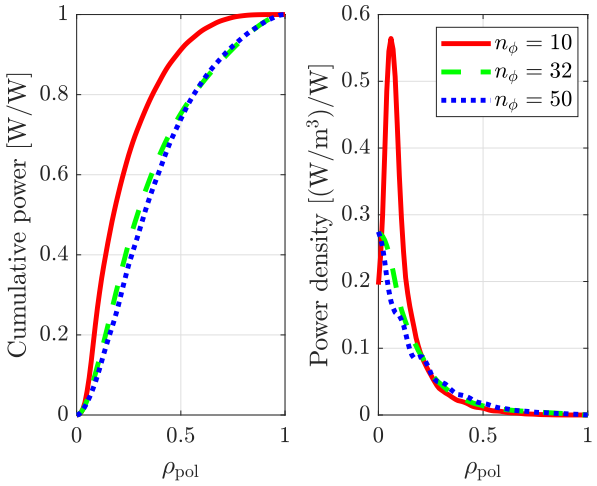


Fig. 15: The left graph shows the normalized amount of power absorbed as a function of ρ_{pol}, $\rho_{\text{pol}} \in [\rho_{\text{axis}}, \rho_{\text{separatrix}}] = [0, 1]$, for $n_\phi = 10$, 32 and 50. The right graph shows power density as a function of ρ_{pol} for $n_\phi = 10$, 32 and 50 and is normalized such that the volume integral of the function will be equal to one.

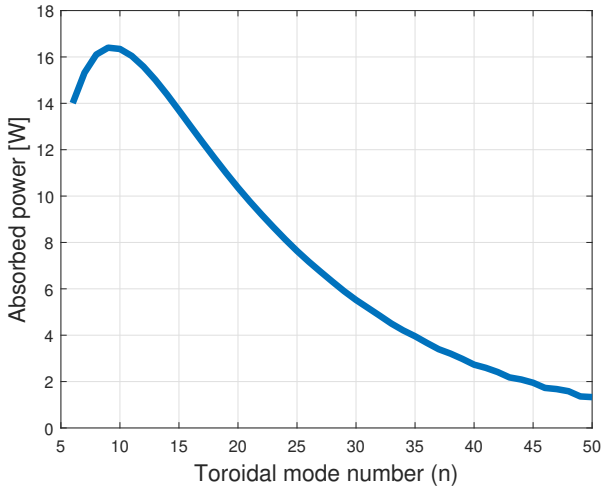


Fig. 16: Total power absorption as a function of n_ϕ in the interval $n_\phi \in [6, 50]$.

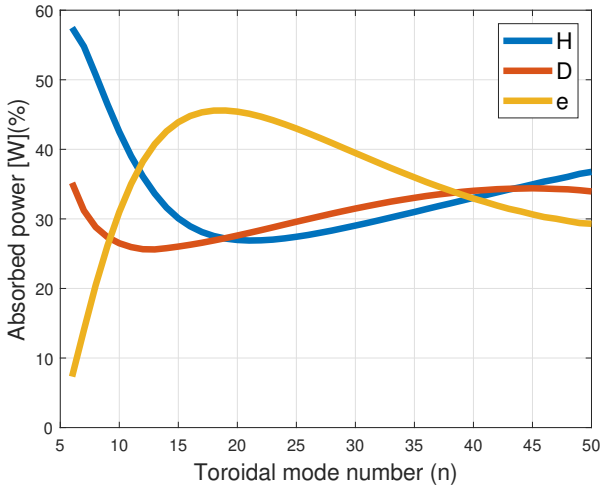


Fig. 17: Partition of power absorption by hydrogen, deuterium and electrons as functions of n_ϕ in the interval $n_\phi \in [6, 50]$.

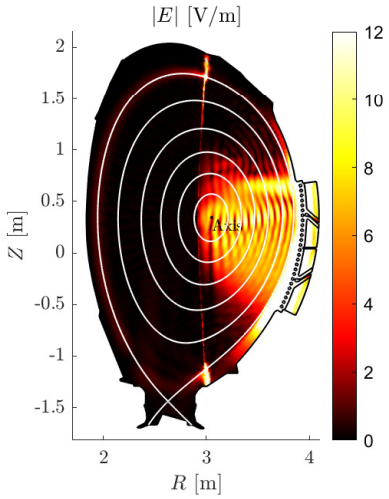


Fig. 18: Electric field strength magnitude plotted for $n_\phi = 8$.

D. Three Dimensional Electric Field

In Fig. 23 a slice of the the real left hand polarized E-field $\text{Re}\{E_+\}$, calculated by using equation (10), in the horizontal plane at $z \approx 0$ and $\phi = [0, 2\pi]$ around the entire reactor can be seen. The electric field was plotted as a Fourier sum for all $n_\phi \in [-50, -6] \cup [6, 50]$. The complete Fourier series runs from $n_\phi = \pm\infty$, but due to limitations of FEMIC and computing power a total of 90 toroidal modes were simulated. The antenna is situated in the 4th quadrant. The electric field magnitude is strongest near the antenna and creates an interference pattern throughout the tokamak. We can see in Fig. 23 that there are 5 wave tops around the plasma in the reactor. This is due to $|n_\phi| \leq 5$ not being included. This is the main reason for errors as the electric field strengths have important contributions for these two modes. There are errors for $|n_\phi| > 50$ that are not negligible as well, as we can see in the SOL where we see a periodicity of about 51. This is again an effect from not including important terms, in this case higher modes than $n_\phi = 50$. We do not however see these higher modes affecting the plasma significantly, due to the waves losing amplitude due to tunneling for these modes.

V. Discussion

A. Dispersion

The dispersive effects in the plasma as have been seen in Fig. 8, show that the there is dampening of the waves concentrated to the center of the plasma. We have seen that absorption is also centered in the plasma in figures 12, 13 and 14, which suggests that the dampening in the dispersion plot in this area comes from absorption. We can also see dampening in other parts of the plasma, but a comparison with absorption plots suggest that this dampening only changes the amplitude of the waves without absorption of the waves. Furthermore we have seen that the wavelengths of the electromagnetic waves

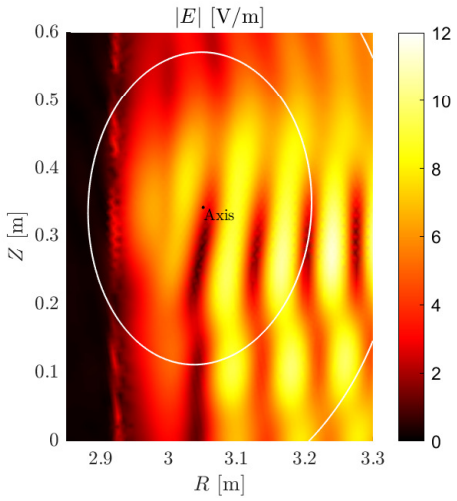


Fig. 19: Electric field strength magnitude for $n_\phi = 8$ around the ion-ion hybrid layer. The ion-ion hybrid layer is shown as the faint line slightly right of $R = 2.9\,\mathrm{m}$. Zoomed in to emphasise where the ion-ion hybrid layer is situated.

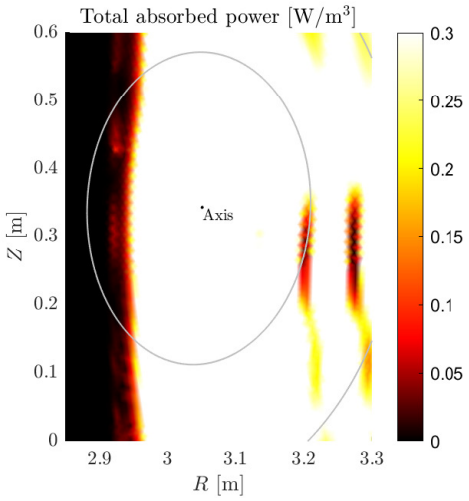


Fig. 21: Power absorption for $n_\phi = 8$ around ion-ion hybrid layer. The ion-ion hybrid layer is shown as the faint line slightly right of $R = 2.9\,\mathrm{m}$. Zoomed in and color scale saturated to emphasize the ion-ion hybrid layer.

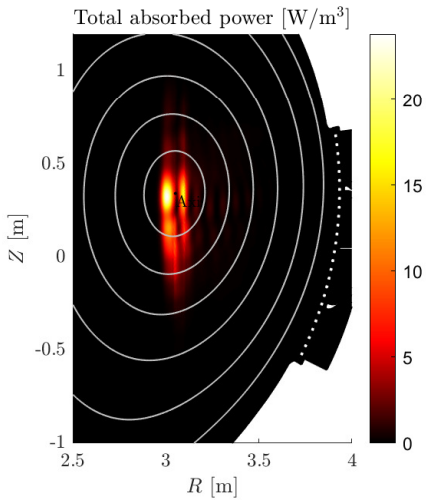


Fig. 20: Power absorption for $n_\phi = 8$.

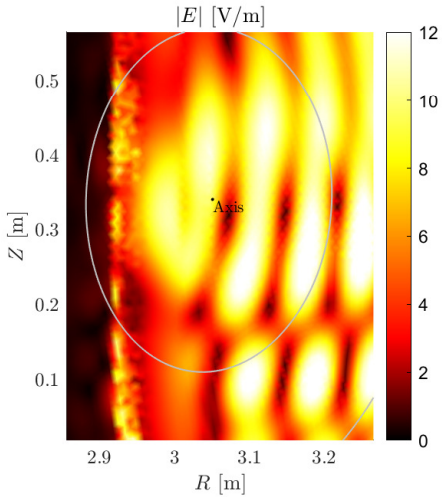


Fig. 22: Electric field strength magnitude for $n_\phi = 5$ around the ion-ion hybrid layer. The plot displays the problems of resolution when simulating for low toroidal mode numbers.

changes throughout the plasma. Most importantly one can notice that the wavelengths becomes significantly shorter around the ion-ion hybrid layer for low toroidal mode numbers. We can conclude that the dispersive effects are most important in the center of the plasma where we want to transfer energy to the plasma from the electromagnetic waves. Lastly, the dispersive effects in the ion-ion hybrid layer have significance due to the singularity in the dispersion relation.

B. Absorption

To measure the quality of heating capability of the plasma we have used a set of parameters:

- Total absorbed power.
- Partition of absorbed power.
- Centering of absorption on ion resonance region.
- The imaginary part of the dispersion relation.

These parameters showed that ICRH efficiently heats the center of the plasma in the desired central regions. While some toroidal modes give less centered absorption, the simulated modes with largest power absorption ($n_\phi \sim [10, 20]$) give a satisfactory absorption in the desired regions, as can be seen for $n_\phi = 10$ in figures 12 and 16. For these modes the total transferred power has a strong dependency on the tunneling phenomenon occurring in the SOL.

C. The Ion-Ion Hybrid Layer

Our analysis shows that the ion-ion hybrid layer has very little effect on the total absorption of the EM waves for $n_\phi \geq 8$. For the toroidal mode numbers simulated, the ion-ion hybrid layer does not approach a singularity. We did not however simulate any scenario for toroidal mode numbers in the interval $|n_\phi| \leq 5$, wherein the singularity in theory should be more prominent. This was due to resolution problems for low toroidal mode numbers, where the electric field solution becomes grainy around the ion-ion hybrid layer, as can be

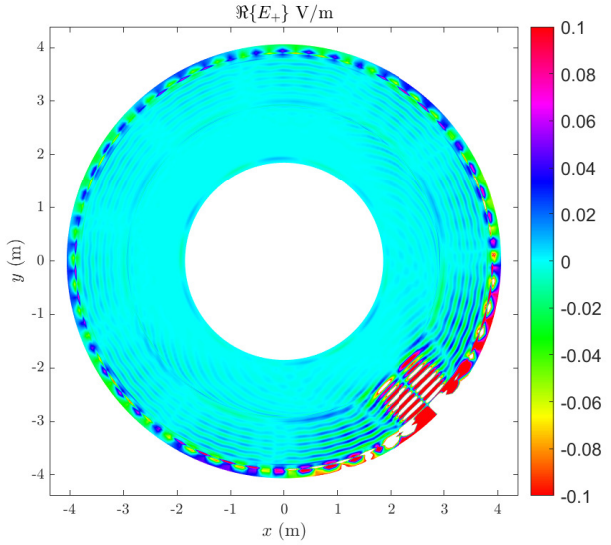


Fig. 23: Real part of E_+, summed over all $n_\phi \in [-50, -6] \cup [6, 50]$.

seen in Fig. 22. Thus, no physical conclusion can be derived. In this regard our analysis is not complete.

Overall our method of deriving conclusions comes from simulations through FEMIC as we have not been able to carry out any real life experiments to collect data. Furthermore, the limitations of time and computer capability have had an effect of us not analyzing the whole spectrum of toroidal modes. This should create some significant discrepancies between ICRH heating and our model. Our results should however shed some light on dispersive effects and heating related to ICRH heating in the JET reactor.

D. Three Dimensional Electric Field

Our last objective was to calculate a 3D electric field to visualize how it spreads out through the whole reactor and analyze the effects from different electric field terms with regard to toroidal mode numbers. The 3D electric field shows a periodicity in the wave patterns. The sum of terms relating to different toroidal mode numbers was not complete leaving some necessary terms out. A periodicity of 5 in the plot suggests that the fields associated to $n_\phi = \pm 5$ have important electric field contributions to cancel out the periodicity in the total field. The field plot also shows a periodicity in the SOL, which means that important terms are left out in the higher modes. Their importance to the electric field inside the plasma however, is not as important as the low modes due to the higher modes tunneling through the SOL. We can also see how the electric field strength is strongest at the antenna and how it dissipates at the resonance region, meaning waves transferring power to the plasma. Furthermore we can notice how the waves reflect on the tokamak walls creating interference patterns throughout the tokamak. The discrepancies of our 3D field and the real field should be major, but our 3D field should give some sense of how the electric field is distributed throughout a slice of the reactor.

VI. Conclusion

The results show that dispersive effects have a dependency on toroidal mode numbers. The amount of power coupled to the plasma from the antenna depends on the exponential decay of the electromagnetic waves in the SOL. The decay of the waves depends on the their wavelength, which in turn depends on toroidal mode number. We have shown that for $n_\phi \leq 10$, there is no tunneling, which means that all power is transferred to the plasma from the antenna. For $n_\phi > 10$ tunneling occurs and decays more with increasing mode numbers.

The aim of ICRH is to have as much power as possible coupled to a central resonance region. In this regard we have shown that both centering of absorption and the width of absorption are better for low toroidal numbers. We have also shown that the electromagnetic fields create standing wave patterns in the plasma. The electric fields are more concentrated to the right of the magnetic axis due to reflections in the ion-ion hybrid layer for low toroidal numbers.

The ion-ion hybrid layer in the dispersion relation for a warm magnetized plasma do not absorb any significant amount of power in relation to resonance regions in our simulations. We did not however include scenarios for $|n_\phi| < 6$. This means that our analysis does not cover the whole spectrum of the electric field. This was the case because of limited computing power and time.

Acknowledgment

The authors would like to thank Thomas Jonsson and Björn Ljungberg for their supervision of this project. They have encouraged and motivated us as well as being engaged and accessible at all times.

References

[1] Robert Rapier. (2020, Jun) Fossil fuels still supply 84 percent of world energy — and other eye openers from bp's annual review. Forbes, Jersey City, New Jersey, US. [Online]. Available: https://www.forbes.com/sites/rrapier/2020/06/20/bp-review-new-highs-in-global-energy-consumption-and-carbon-emissions-in-2019/?sh=4aa6af5b66a1

[2] M. Widell, F. Björkenwall, M. Kryssare. (2020, Aug) Hybrit: Ssab, lkab och vattenfall startar världens första pilotanläggning för fossilfritt stål. Vattenfall, Stockholm, Sweden. [Online]. Available: https://group.vattenfall.com/se/nyheter-och-press/pressmeddelanden/2020/hybrit-ssab-lkab-och-vattenfall-startar-varldens-forsta-pilotanlaggning-for-fossilfritt-stal

[3] J. P. Freidberg, *Plasma physics and fusion energy*. Cambridge, England: Cambridge University Press, 2007.

[4] P. A. Vallejos Olivares, "Modeling rf waves in hot plasmas using the finite element method and wavelet decomposition : Theory and applications for ion cyclotron resonance heating in toroidal plasmas," Ph.D. dissertation, Kungliga Tekniska Högskolan, Dec 2019.

[5] Kirsten Haupt. (2018, Jun) Searching for the perfect shape. ITER ORGANIZATION, Saint-Paul-lez-Durance, France. [Online]. Available: https://www.iter.org/newsline/-/3037

[6] I. P. B. Editors, I. P. E. G. C. an Co-Chairs, I. J. C. Team, and P. Unit, "Chapter 1: Overview and summary," *Nuclear Fusion*, vol. 39, no. 12, pp. 2137–2174, Dec 1999. [Online]. Available: https://doi.org/10.1088/0029-5515/39/12/301

[7] Fusionwiki. (2015, Apr) Toroidal coordinates. [Online]. Available: http://fusionwiki.ciemat.es/wiki/Toroidal_coordinates

[8] T. Jonsson, "Electromagnetic waves in dispersive media," KTH, Stockholm, Sweden, 2021, lectures notes from Kungliga Tekniska Högskolan.
[9] D. G. Swanson, "Plasma waves (2nd edition)," *Plasma Physics and Controlled Fusion*, vol. 45, no. 6, May 2003. [Online]. Available: https://doi.org/10.1088/0741-3335/45/6/701
[10] MATLAB, *R2020b*. Natick, Massachusetts: The MathWorks Inc., 2020.
[11] R. R. T. H. L. F. Pablo Vallejos, Thomas Jonsson, "Effect of poloidal phasing on ion cyclotron resonance heating power absorption," *Nuclear Fusion*, vol. 59, no. 7, Jul. 2019.
[12] C. Multiphysics, "Introduction to comsol multiphysics®," *COMSOL Multiphysics, Burlington, MA*, vol. 9, p. 2018, 1998.

CONTEXT H
OBSERVATIONS IN SPACE

POPULAR DESCRIPTION

Electrons and moons: pieces of the same puzzle

Have you ever looked at the night sky and wondered what was out there? Or what you are made of? Elementary particles and moons might seem different but they are all part of the same grand design. Moons are made of particles and the particles in turn are affected by the gravitational pull of the moons, to gain a complete picture of the world we need to understand events on all scales. One way to acquire this picture is to take a closer look at grand events taking place in space.

Observing events in space is no easy task! Until recently we could only look at these objects from the surface of the Earth with our eyes or with rudimentary land based telescopes. This is no longer the case since in the last century, mankind has made great strides in this area. We are now capable of building satellites and telescopes that peer through the vastness of space. With these instruments, we can look for both water on the moons of Jupiter and track the velocity of almost unperceivable particles called electrons zipping around in the magnetic fields around Earth.

These discoveries in space will have a huge impact on our ability to one day take a step beyond our own planet. In a couple of decades we might send humans to Mars, instead of rovers. If we do this, we want to have the best available information about the conditions of space. Theoretical knowledge has taken us far, but there is much to be gained from actual observations. Further in the future we might establish space stations around the outer planets like Jupiter or Saturn in order to observe them more closely. But before then, it is crucial that we carry on our search for knowledge through observations of space, in order to pave the way for our future as a multi planetary species.

SUMMARY OF PROJECT RESULTS

Space physics covers the area in physics that deals with the environment and phenomena occurring in the solar system. Examples of these environments are the sun, the planets, moons and asteroids. Space physics also encompasses how these objects interact with each other, for instance how the particles and radiation from the sun affects the other bodies in the solar system. These phenomena are usually studied with help of satellites orbiting around the Earth or other bodies in space and by the use of telescopes, which can both be placed on the Earth's surface or in space.

Using data from two different NASA missions, Hubble Space Telescope and the Magnetospheric Multiscale Mission (MMS), it is possible to probe events and phenomena both nearby and far away. MMS consists of four satellites orbiting in a formation around the Earth, gathering precise data of the charged gas in short bursts. Similar conditions as those near Earth occur during grand cosmic events such as supernovas, and so studying the microphysics in the near-Earth space will lead to a better understanding of these complex events as well. The Hubble Space Telescope on the other hand has taken high definition images of far away objects such as Jupiter and its moons, allowing for a closer look at the environment of the moon p. Knowing the conditions on Europa can reward us with a better estimation as to the conditions on planets and moons which are too far to observe in any greater detail with modern technology.

The group tasked with project H2 set out to gather more information about the conditions in bow shocks forming the Earth's magnetic field. These shocks form in many different places in the universe, and are highly energetic and complex, and lead to acceleration of charged particles. Normally acceleration in a shock is exacerbated by collisions between particles or molecules, but the conditions in space hinder such collisions. By gathering data about the density of electrons with certain energies whilst the satellites are crossing the bow shock, it is possible to rank the acceleration and cross-reference with other gathered variables. In order to better understand what is causing these collisionless shocks the group H2 set out to create a

ranked list of the burst events, in order of most energetic to least, and possibly categorize each entry by the angle between the acceleration direction and the magnetic field. That way, future groups may use the list as a reference or inspiration to create a better theory as to how these acceleration events occur. This list can always be expanded in the future by correlating more variables or increasing the precision of the ranking system, as well as using data from more events captured by the MMS satellites.

Project group H3 has with help of data from the Hubble telescope reanalyzed images that identified anomalies as possible evidence of water plume activity on Jupiter's moon Europa. With a given algorithm, systematic uncertainties and statistics around the limb have been investigated in order to determine whether such plumes exist. The results from this study will be interesting and important for two future space missions, ESA's Jupiter Icy moon explorer (JUICE) and Europa Clipper missions, since the occurrence of plumes causes a constraint for planning the scientific measurements that will be done by the science instruments.

IMPACT ON SOCIETY AND ENVIRONMENT

Observations in space have relatively little direct impact on society. For example it is not always clear what the immediate gain is from observing far away moons or measuring the acceleration of electrons in space, other than a general gain of knowledge. However, the gaining of knowledge is not something to discard, since it may have an effect on the quality of life of mankind, both in terms of technological advancements but also to satisfy our will to understand nature and where we come from. Most modern technology is based on knowledge of physics that was acquired many decades if not centuries ago, like e.g. quantum physics, which is heavily used in transistor circuits and in the development of LEDs.

Environmental effects might not be a direct effect from the two projects, but to make the observations and gather data, instruments are required. The instruments used in the projects are sent to an Earth orbit or fly further and contribute to the clutter of space. Ideally, there is a contingency plan for when the instruments are obsolete, but in the worst case they are left in a descending orbit with a risk of colliding with other satellites which further worsens the issue. Without a set plan there might be a time in the future where certain orbits are so occupied that it will be hard to send out more. On the bright side though, the instruments used in these projects worsen this clutter significantly less than gps satellites or other low Earth orbit satellites, since they have specific locations they need to measure far away from low Earth orbiting satellites.

Another aspect to consider is whether it is justified to spend resources on the exploration of space when there still exists a large amount of issues here on Earth today like global warming, famine etc. that are much more important. The argument can be made that space exploration will not lead to a lot of direct improvements for the general well being of ordinary humans. If the money for instance had been spent on making sure some people got access to clean water, it would most certainly directly improve the life standard for some people. On the other hand, the amount of resources being devoted to space research is merely a small fraction compared to other fields and departments. One notable example is military research and general military spending.

One socially beneficial aspect of space research that should not be forgotten is the fact that the field often incentivizes or leads to cooperation between different nations. This can contribute to global stability, which in the long term will lead to better lives for all people.

Electron Acceleration at Earth Bow Shock

Karl Bergson Hallberg

***Abstract*—Electrons accelerating in shock events occur often in outer space, for example in supernovas and the poles of black holes, and are of high interest to physicists and other researchers. The technology to take a closer look at events that far away does not exist yet, but luckily we can observe similar events in places where the Earth's magnetic fields and particle streams from the Sun meet. Using data from NASA's MMS mission this paper aims to gather information about what variables affect the acceleration, under what conditions the most energetic events occur and create a ranked list of several hundreds of these events. It did this by calculating the expected value of the electron distribution function at different times to create a dimensionless ranking. The study showed that these events are highly complex and that it is difficult to assign a few variables which would affect the acceleration. However it also showed that most acceleration occurs after the most abrupt shock crossing and not exactly at the location where the expected value is maximal, and that there are some correlations with angle relative to the solar magnetic field and electron number density.**

***Sammanfattning*—Elektroner som accelereras i shockar sker ofta i yttre rymden, till exmepel i supernovor och vid polerna hos svarta hål, och är därför av högt intresse hos fysiker och andra forskare. Teknologin för att titta närmre på dessa fjärran fenomen existerar inte ännu, men som tur är så kan vi observera liknande händelser på platser där Jordens magnetfält möter partikelvindar från Solen. Med hjälp av data från NASAs MMS uppdrag har detta projekt önskat att samla information om vilka variabler som påverkar accelerationen, under vilka omständigheter de mest energirika händelserna sker och skapa en rankad lista av flera hundra av dessa händelser. Det gjorde detta genom att beräkna det förväntade värdet på elektronernas distributionsfunktion vid flera tillfällen för att skapa en dimensionslös rank. Studien visade att dessa händelser är mycket komplexa och att det är svårt att tilldela ett fåtal variabler som skulle påverka accelerationen. Dock så visade projektet att den största accelerationen sker efter den mest abrubta shockkorsningen och inte exakt vid det tillfälle då det förväntansvärdet är som högst, och att det finns någon korrelation med vinkeln relativt solens magnetfält och elektronernas nummerdensitet.**

***Index Terms*—electrons, acceleration, bow shock, space, MMS, IRFU**

Supervisors: Andris Vaivads, Martin Lindberg

TRITA number: TRITA-EECS-EX-2021:163

I. Introduction

We observe many events of high energetic particle acceleration and plasma heating forming in various astrophysical shock events in the universe, leading to the creation of radio waves and X-rays which we can observe on astronomical distances. These extreme events are complex and non-linear which lead to difficulties understanding the phenomena involved. In particular, electrons are accelerated to non-thermal energies and then participate in other astrophysical mechanisms such as diffusive shock acceleration of cosmic rays. However the exact

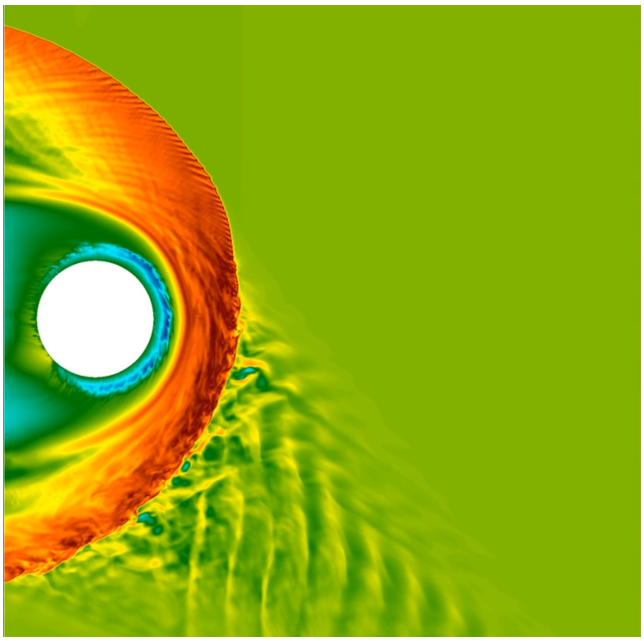


Fig. 1. Still image from a vlasiator simulation of the solar wind meeting Earth's magnetic field. Image courtey of Helsinki University Vlasiator simulation [1]

way in which the acceleration takes place is not completely understood. [2], [3]. To gain a better picture of how these events can take place involve getting a better understanding of the collisionless shocks accelerating these electrons [4]. NASA's MMS mission (start 2015) allow for a high resolution look at the closest astrophysical shock, the earth bow shock, in which particles ejected from the sun meets the magnetic field of earth. Using the instrument suite SMART (Solving Magnetospheric Acceleration, Reconnection and Turbulence) on these four satellites we have access to data from many of the high acceleration events forming in the shock [5].

Many singular events have already been analyzed thoroughly (see e.g Chen et. al., Oka et. al.) but there is a lack of statistical analysis of many events. Chen et. al. and Oka et. al looked at the heating and acceleration of electrons in singular events, where and when the acceleration takes place, where the electrons gain the highest thermal energy and what processes would lead to this. Since many theories as to why this acceleration occurs has been proposed, the papers have looked at an event and tried to come up with a mechanical reason the acceleration occurred in that specific event. They have also proposed new mechanics which could potentially explain the phenomena and tried to exclude reasons that would not.

The magnetic field lines of Earth would if undisturbed look like those of a dipole, however solar wind consisting

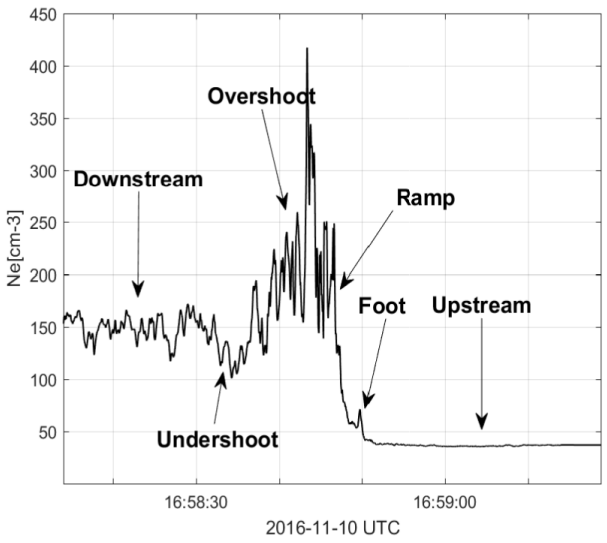


Fig. 2. The shock crossing from a density perspective, showing the different stages of the satellites crossing the shock. Image courtesy of M. Svensson [6]

of charged particles disrupt this by having their own magnetic fields and which changes the shape. In Figure 1 a Helsinki University Vlasiator simulation of the solar wind meeting the magnetic field is shown. The angle between the normal of the bow shock and the direction of the solar wind magnetic fields are an important factor when looking at events. The shocks where the solar wind magnetic field is approximately parallel to the shock normal are called quasi-parallel shocks and the other are called quasi-perpendicular shocks. The quasi-perpendicular events are much more structured and easier to identify, but the quasi-parallel often times involve higher acceleration and heating. As seen in the picture there is a calm region with a clean line showing the bow shock near the top of the image which is characteristic of quasi-perpendicular shocks, and a much more chaotic region at the bottom where it is quasi-parallel.

Since much is known about singular events, we decided to further the understanding by calculating values which are easily obtained and can be done en masse. By targeting the easier but still important values we hope to further the scientific understanding and give clues or eliminate some theories as to why it is happening. Specifically we want to find out where in the shocks the electrons are heated up the most, where the highest acceleration occurs and correlate this with the solar wind magnetic field angle relative to the shock normal. We aim to create a ranking system, based on acceleration, and order the events we look at from most to least energetic. Also correlate this with a calculated angle between the shock normal and solar winds and draw conclusions based on these statistics.

II. Work

A. Method

The MMS (Magnetospheric MultiScale) mission consist of four satellites flying in a variable close tetrahedron formation, orbiting the Earth in an oval shape as seen in Figure 3. The four satellites are equipped with various instruments to measure amongst other things the electrical and magnetic fields, particle velocities and particle densities. The satellites spin at a rate of 3 RPM during operation and have deployable booms in order to measure the fields in each direction, at a distance larger than the physical size of the satellite [7]. Not all instruments on the spacecraft are used in this report, which uses data mainly from the FIELDS investigation and the FPI (Fast Plasma Investigation) suites.

The FIELDS suite includes the deployable booms and consists of two axial and four spin-plane electric field sensors, two flux-gate magnetometers, a search-coil magnetometer, and two electron drift instrument per MMS spacecraft. The FPI suite on the other hand includes four electron and ion spectrometers. When the data from the these spectrometers are combined a velocity-space distribution of the electrons from 10 eV to 30 keV with a time resolution of 30 ms is achieved [5]. The spacecrafts orbits the Earth and crosses the bow shock lines as seen in Figure 4. At certain times in the orbit the satellites gather data in the so called burst mode, where it captures and sends detailed information from all instruments. We use data from this mode taken during a shock crossing to do our calculations.

The two types of shocks explained earlier, quasi-perpendicular and quasi-parallel, have distinct forms and play a large role in how structured and easily recognized a shock crossing is. A quasi-parallel shock is one where the magnetic field of the solar wind is angled between 0 and 45 degrees in reference to the bow shock normal, while quasi-perpendicular shocks are between 45 and 90 degrees. In Figure 1 a simulation of the solar wind magnetic field meeting the Earth's can be seen, where the lower half of the picture shows a quasi-parallel shock and the upper half a quasi-perpendicular shock. Quasi-parallel shocks are generally more chaotic and complex which hinders our ability to identify the times where the satellites are crossing the bow shock. A typical quasi-perpendicular shock crossing is seen in the density plot of Figure 2. It shows the stable region of the solar wind (upstream) where the density is relatively low and unchanging and the sudden spike corresponding to the time of crossing leading to the more unstable magnetosheath region (downstream). At the time of crossing the density increases rapidly for quasi-perpendicular shocks which makes for an easy marker for these shock crossings.

The Swedish Institute for Space Physics at Uppsala Universitet (IRFU) has done much work in creating data science tools to process and interpret the data gathered by the satellites. This includes tools to import the data and plot it for interpretation. Ahmad Lalti of IRFU has created a list of many events using a neural network. This list contain a total of 556 events recorded from October 2015 until January 2016. These events contain 143 burst mode events and 136 of the burst mode events are actual shock crossings. These events were downloaded and used in our calculations. This list is AI generated and therefore contains time intervals which are not always optimized for our purposes, for example the shock crossings and times of interest are in reality only about a minute long but the unaltered events

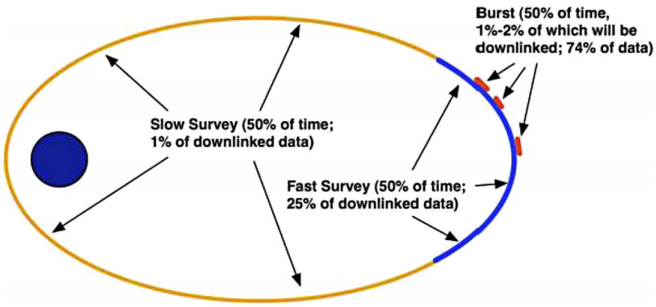


Fig. 3. Orbits of the MMS satellites around Earth, showing the locations of burst data gathering where our data is taken from. Image from NASA [7]

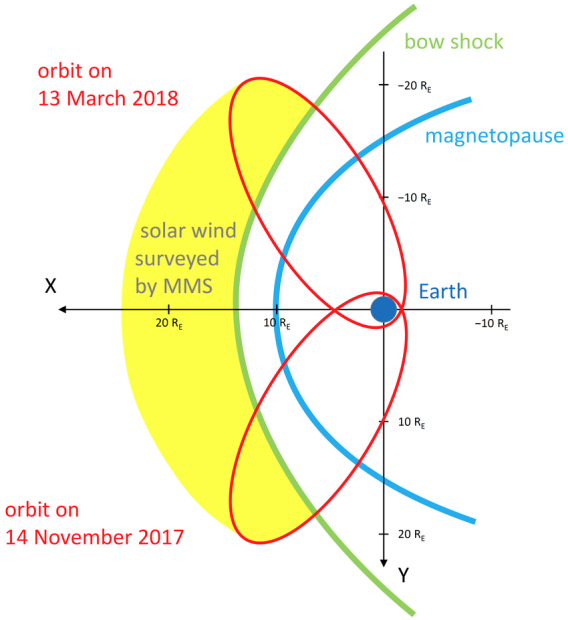


Fig. 4. Orbits of the MMS satellites, showing the crossing of the bow shock. Image from Ferdinand Plaschke [8]

could be more than two minutes surrounding the crossing. We use the functions from IRFU to model an approximation of the angle of the solar wind magnetic field relative to the shock normal by inputting local magnetic field values and solar wind data from a satellite at Lagrange point L1.

When the shock crossing occur an increase in electron number density is seen. This density increase leads to a general heating of the electrons which increases the average velocity, and is simply called heating. We are searching for acceleration out of the ordinary and therefore need to take this into account when doing calculations. The satellites provide data in the form of the electron density distribution function, which gives the phase-space density of the electrons as a function of energy and time. When heating occurs the shape of the distribution functions flattens and the average velocity increases, which we use as another marker for the heating. Since the heating is associated with increased energy and higher velocities, we center our search around it. We alter the time intervals of the events by centering them around the time where the number density was at its maximum and using data only within 30 seconds of this time.

The electron number density is a good marker for the crossing for quasi-perpendicular conditions. But in quasi-parallel conditions the number density does not always decrease after the crossing, as can be seen in plot number two in Figure 5 where it continues to rise and fall throughout the time interval. Therefore we calculate the expected value of the electron density distribution function and use its time of maximum as a marker for the crossing. This marker is more resistant to sudden changes and still work as intended for quasi-perpendicular events. With it we capture what energy the electrons are most likely to occupy at any given moment and approximate well where the heating occurs. The distribution function is shown in the logarithmic plot at the bottom of the figure and the black line marks the time where it has its maximum. We calculate the expected value by using the formula

$$E_{exp}(t) = \int\int\int E(t) f(E(t), t, \theta, \phi) dE d\theta d\phi, \qquad (1)$$

and integrating over all energies and angles collected by the satellites. The surrounding conditions are harsh and produce noise. To not skew the expected value by including low-energy electrons from the satellites and random high-energy electrons which can influence the distribution function we only integrate over energy values ranging from 30 eV up to 1 keV.

We are looking for unusual acceleration of the electrons, and decided to quantify this by referencing the most common energy value for the electrons. A sudden increase of electrons with an energy well above the expected value points to a possible unusual acceleration. The distribution function normally decreases monotonically at higher energy, but a sudden acceleration will cause this distribution to increase suddenly at energies most commonly around or above 1 keV. This corresponds to a tail in the distribution function where we see it decrease and then a spike appears at higher energies.

To find these tails, we use the calculated expected value and find the times where the distribution function shows a large density of electrons at higher energies. At ten times larger energies than the expected value, we find that the values are large enough that we can easily identify abnormal spikes without going to even larger energies where noise may begin to be a much larger factor. We want to compare acceleration of one event to others and need a number which is independent from fluctuating values such as expected value and temperature of the surroundings. A dimensionless number we can calculate for each event is the ratio of electron phase-space density at energies ten times the expected value with the phase space density of electrons at energies around the expected value. By dividing the two phase-space density values at the time of acceleration, we get the dimensionless number which quantify the acceleration. A shock crossing which has a large acceleration will have a large ratio of density of high energy electrons compared to the lower energies, and vice versa. In addition, we saved the expected value, angle of the solar wind magnetic field, electric and magnetic field values and electron number densities for each event so that we can compare across events.

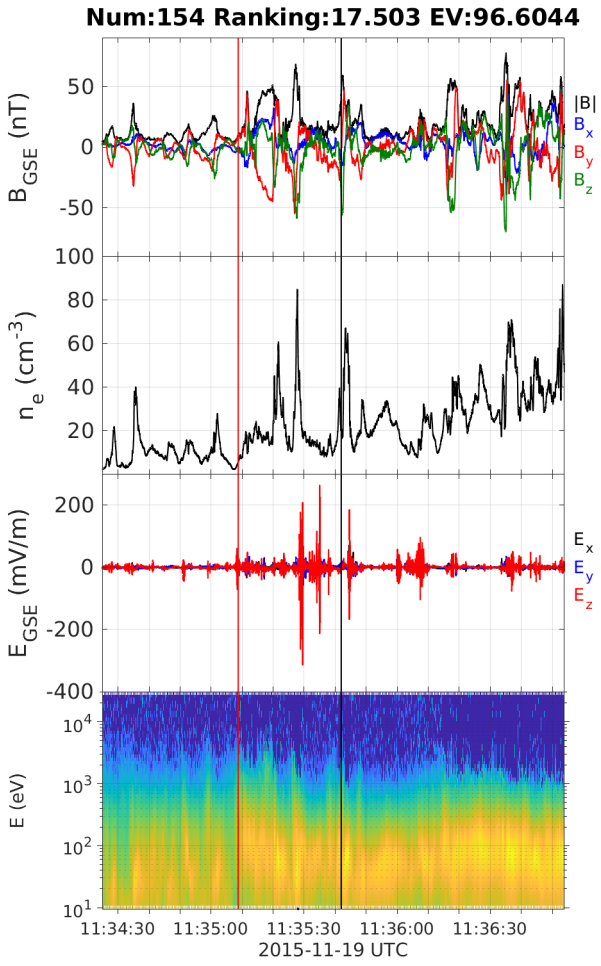


Fig. 5. Highest ranking event, which has the characteristics of a quasi-parallel shock crossing as seen from the unstructured and unstable number density and distribution function plots.

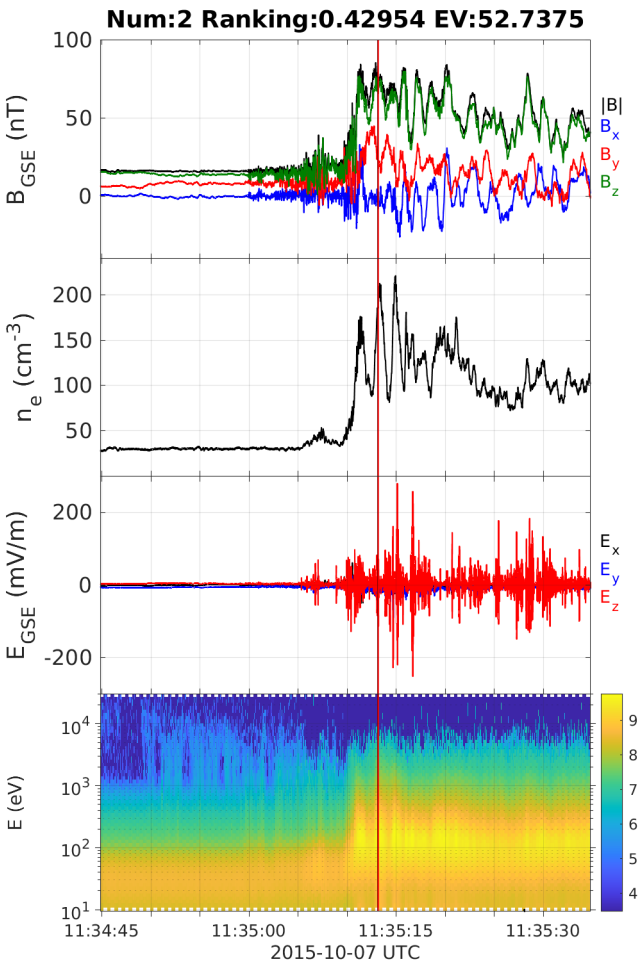


Fig. 6. A typical quasi-perpendicular shock showing the satellites going from the stable magnetosphere region, into the shock crossing and the solar wind

B. Results

Out of the 556 total time spans provided, a ranked list of 146 events from October 2015 to January 2016 was created successfully. The ranking numbers spanned from 17.5 for the highest ranked event down to 0.061 for the lowest. A ranking number above 10 was rare and only four events ranked higher. The expected value of the heating was saved for all of these events and spanned from 138.5 (eV) to 89.3 (eV). Angle approximation produced a number for 115 of the 146 events, with a maximal angle of 89.3 and minimum of 0. Linear fit on the angle approximation on the ranked list revealed a small correlation (inclination of -0.092) but the variation was very large as can be seen in Figure 7. We saved the time of heating and time of acceleration for all of these events, which is marked by red and black lines (respectively) on the plots. In the majority of cases the time of heating occured for quasi-perpendicular events at or shortly after the shock crossing as expected and the acceleration at the same time or shortly after the heating.

By physical inspection we observed that most of the highest ranked events had the form of a quasi-parallel crossing, even when the approximated angle indicated otherwise. As an example event number 154 seen in Figure 5 shows the highest ranked event which has the characteristics of a quasi-parallel event (no clear shock crossing, unstable density and energy spectrogram and small differences between upstream and downstream). The red line marks the time of maximal expected value and the black line the time of acceleration. The density is increasing with time with is suggestive of a shock crossing and the time of acceleration occurs after the heating which is in line with other events. The first clear quasi-perpendicular crossing was number 19 in the list and had a ranking number of 2.1, about 8 times smaller than the highest ranked event. The eight lowest ranked events may be quasi-parallel since they do not have the form of a quasi-perpendicular event, but it is difficult to determine if quasi-parallel events are captured during crossings or simply data gathered in the magnetosheath. These eight events have ranking number spanning from 0.082 down to 0.061.

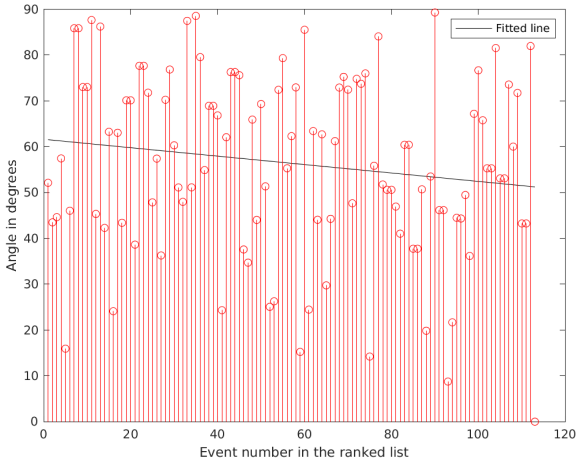


Fig. 7. Plot of the successfully approximated angles for each event in the ranked list, showing low correlation and a linear fit with a coefficient of -0.092

III. Discussion and Conclusion

The result of the data gathering has been both as expected and a bit surprising. In the events that are clear and have the classic look of a quasi-perpendicular shock crossing, such as in Figure 6, the calculated values and locations are fairly accurate as to our guesses to where they would be. Unfortunately not all events are of the same type. An example of this can be seen in Figure 5, in which the chaotic structure of a quasi-parallel shock crossing is seen. Events such as those generally had a very high ranking number, often ten or twenty times the other events, pointing to the complex interactions involved in the shocks. One of the highest ranking quasi-perpendicular shock crossings was number 139 in the list. In event number 139, in Figure 8, the time of acceleration and maximum expected value are very close, though they are not exactly at the crossing. This is common for quasi-perpendicular events, but for quasi-parallel they are often separated by 30 seconds or more. The expected value in the solar wind is however not very high, which makes the high value of the acceleration more interesting.

The ranking system seems sound. This can be seen by looking at events which are both high and low ranked, for example event number 2 which is the lowest ranked event, and is shown in Figure 6. This shock crossing has one of the lowest expected values which gives credence to the ranked list. One result which also was not expected was how relatively uncorrelated the location of acceleration were to the electric and magnetic fields, rarely occurring at the times of maximal field strength. At the beginning of the work, we had thought that many of the calculated values (such as electron number density and electromagnetic field strengths) would be highly correlated, but this did not turn out to be the case. Even the angle between the shock normal and the solar wind magnetic fields were not very correlated, often varying between similarly ranked events. Although the calculated angle was a rough approximation, but we expected a stronger trend.

A problem we encountered was quality and quantity of

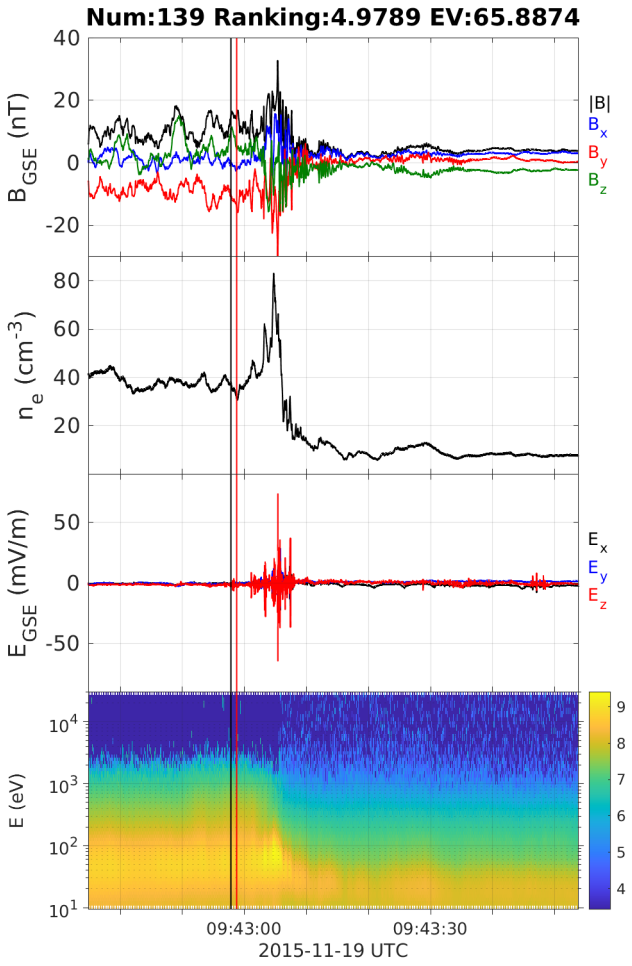


Fig. 8. A very high ranking quasi-perpendicular shock crossing. Time of heating and acceleration are close which is typical of quasi-perpendicular shocks. The expected value is fairly low which makes the high acceleration interesting.

the events in question, the list we used was generated by an AI and therefore not manually checked. Therefore a lot of the events were not optimal for the sorts of calculations we decided to do, their usefulness rises the more clean the event is. In addition many of the time periods in the list were not of burst events, making them unusable for us. In the end only about about a fifth of all the time periods in the list were burst events, and some of those were not of shock crossings. The events which are clearly depicting a quasi-perpendicular shock crossing show results similar to those expected; the heating occurs exactly at or very close to the crossing and the acceleration follows thereafter.

In conclusion, we managed to finish our goal of creating a ranked list of all the events provided in the list and correlate with solar wind magnetic field angle. We did not manage to do much analyzing of the mechanics involved or look closer at singular events, but as mentioned that has been done by several other groups. We hope that some other group has a use of our list and saved values and finds it helpful in narrowing down the reasons this acceleration occurs.

APPENDIX

A. Table of the ten Highest Ranked Events

Event Number	Ranking Number	Angle (degrees)	Maximum Expected Value (eV)
154	17.50	NaN	138.45
30	10.53	NaN	53.89
161	8.83	52.13	106.66
130	8.70	43.50	82.93
549	8.52	44.64	102.79
118	6.90	57.45	150.8
556	6.84	15.94	69.19
147	6.83	52.85	68.35
139	4.98	46.03	83.78
94	3.65	85.85	83.78

B. Plots and Code Generated During the Project

The GitHub link below contains all MATLAB the code used in the thesis work and the final ranked list. In addition, the second link contains the plots of the electric and magnetic fields, electron number density and spectrogram of the distribution function made for all the events in this thesis work.

- https://github.com/EttNollEtt/H2-Electron-Acceleration-at-Bow-Shock
- https://kth.app.box.com/v/KEX2021KBHallberg

ACKNOWLEDGMENT

The authors would like to thank supervisors Andris Vaivads and Martin Lindberg, Ahmad Lalti and the team at IRFU.

REFERENCES

[1] "Simulations", *Vlasiator, University of Helsinki*. April, 2021. [Online]. Available: https://www2.helsinki.fi/en/researchgroups/vlasiator/simulations

[2] M. Oka et. al., "Electron Scattering by High-frequency Whistler Waves at Earth's Bow Shock", *The Astrophysical Journal Letters*, vol. 842, no. 2, June 20. [Online]. Available: doi: 10.3847/2041-8213/aa7759

[3] T. Katou and T. Amano, "Theory of Stochastic Shock Drift Acceleration for Electrons in the Shock Transition Region", *The Astrophysical Journal*, vol. 874, no. 2, April 2019. [Online]. Available: doi: 10.3847/1538-4357/ab0d8a

[4] L.-J. Chen et. al., "Electron Bulk Acceleration and Thermalization at Earth's Quasiperpendicular Bow Shock", *Physical Review Letters*, vol. 120, no. 22, May 2018. [Online]. Available: doi: 10.1103/PhysRevLett.120.225101

[5] D.N. Baker et. al., "Magnetospheric Multiscale Instrument Suite Operations and Data System", *Space Science Review*, vol. 199, pp. 545-575, Feb 2014. [Online]. Available: doi: 10.1007/s11214-014-0128-5

[6] M. Svensson. "Electron heating in collisionless shocks observed by the MMS spacecraft" M.S. Thesis. Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology. Luleå, Sweden. March 2018.

[7] "Unlocking the Secrets of the Electron Diffusion Region" D. Smith. NASA. May 2021. [Online]. Available: https://mms.gsfc.nasa.gov/

[8] P. Ferdinand. "How much solar wind data are sufficient for accurate fluxgate magnetometer offset determinations?" *Geoscientific Instrumentation, Methods and Data Systems Discussions.* 1-11. Mar 2019. [Online]. Available: doi: 10.5194/gi-2019-4.

Search for Water Plumes on Jupiter's Moon Europa

Ebba Blom and Wilhelm Branner

***Abstract*—Several studies have been conducted with the aim of finding water plumes at Jupiter's moon Europa. The possible evidence of plumes is important for two future space missions, the JUpiter ICy moons Explorer (JUICE) and the Europa clipper mission. Multiple observations of Europa in transit of Jupiter have been obtained with the Hubble Space Telescope (HST). In an analysis by Sparks et al. 2016 three plume candidates were found. Later on, in the report Sparks et al. 2017 an additional candidate was found at a similar location as a previous candidate, potentially making the evidence of plume existence stronger. In 2020 Giono et al. reproduced the results from Sparks et al. 2016 and found uncertainties in the method that had been used when finding these plume candidates. First Giono et al. claim that the position of Europa in the observation was inadequately determined. Also, positive outliers had not been considered when analyzing the z-statistic. In this study, the algorithm developed by Giono et al. has been used to reproduce the results from Sparks et al. 2017 where additional evidence for plume activity was presented. The algorithm was then applied to the previous observation, where plume activity had been found in the same location. Lastly, the two observations were merged into one image and the algorithm was applied once again. The results of the z-statistic from the observations gave large negative outliers which can be considered as plumes. However, positive and negative outliers had similar significance for the two independent observations which somewhat diminishes the evidence. Also, misalignment between model and observation generates distorted statistics. The statistical uncertainties and fluctuations can easily be mistaken as evidence of plume existence.**

***Sammanfattning*—Flera studier har utförts med syftet att upptäcka vattenplymer på Jupiters måne Europa. Det eventuella beviset för att plymer existerar är viktigt för två framtida rymduppdrag, JUpiter ICy moons Explorer (JUICE) och Europa clipper mission. Flera observationer av när Europa passerar framför Jupiter har erhållits av Hubbleteleskopet (HST). I en analys av Sparks et al. 2016 har tre kandidater för plymer hittats. I rapporten av Sparks et al. 2017 hittades ytterligare en kandidat på ungefär samma ställe, vilket potentiellt kunde stärka beviset för att det var plymer. År 2020 reproducerade Giono et al resultaten från Sparks et al. 2016 och hittade brister i metoden när dessa plymkandidater hittades. Först poängterar de att Europas position i observationen var felaktig. Dessutom togs det inte hänsyn till positiva avvikelser när z-statistiken betraktades. I den här rapporten har algoritmen som skapades av Giono et al. använts för att reproducera resultaten från Sparks et al. 2017 där ytterligare bevis för plymer presenterades. Algoritmen applicerades sedan på den föregående observationen där bevis för plymer hade hittats i samma område. Därefter slogs dessa bilder ihop och analyserades som en bild. Resultatet av z-statistiken från observationerna gav stora negativa avvikelser vilket kan ses som en plym av vattenånga. Dock eftersom positiva och negativa avvikelser hade liknande signifikans för de två enskilda observationerna försvagas beviset för att plymer skulle existera. Dessutom genererar lokala fel mellan modell och observation en förvrängd statistik där statistiska osäkerheter och fluktuationer enkelt kan misstas som bevis för att vattenplymer existerar.**

***Index Terms*—Europa, Jupiter, Hubble Space Telescope.**

Supervisor: Lorenz Roth

TRITA number: TRITA-EECS-EX-2021:164

I. Introduction

The icy surface of Jupiter's moon Europa is relatively young which indicates some geological activity, the moon is known to have a large underground ocean and therefore it has long been speculated that cryovolcanism is a regular occurrence on the moon. This would lead to large water plumes above the surface, similar to the ones at Saturn's moon Enceladus which were detected by the Cassini spacecraft [1].

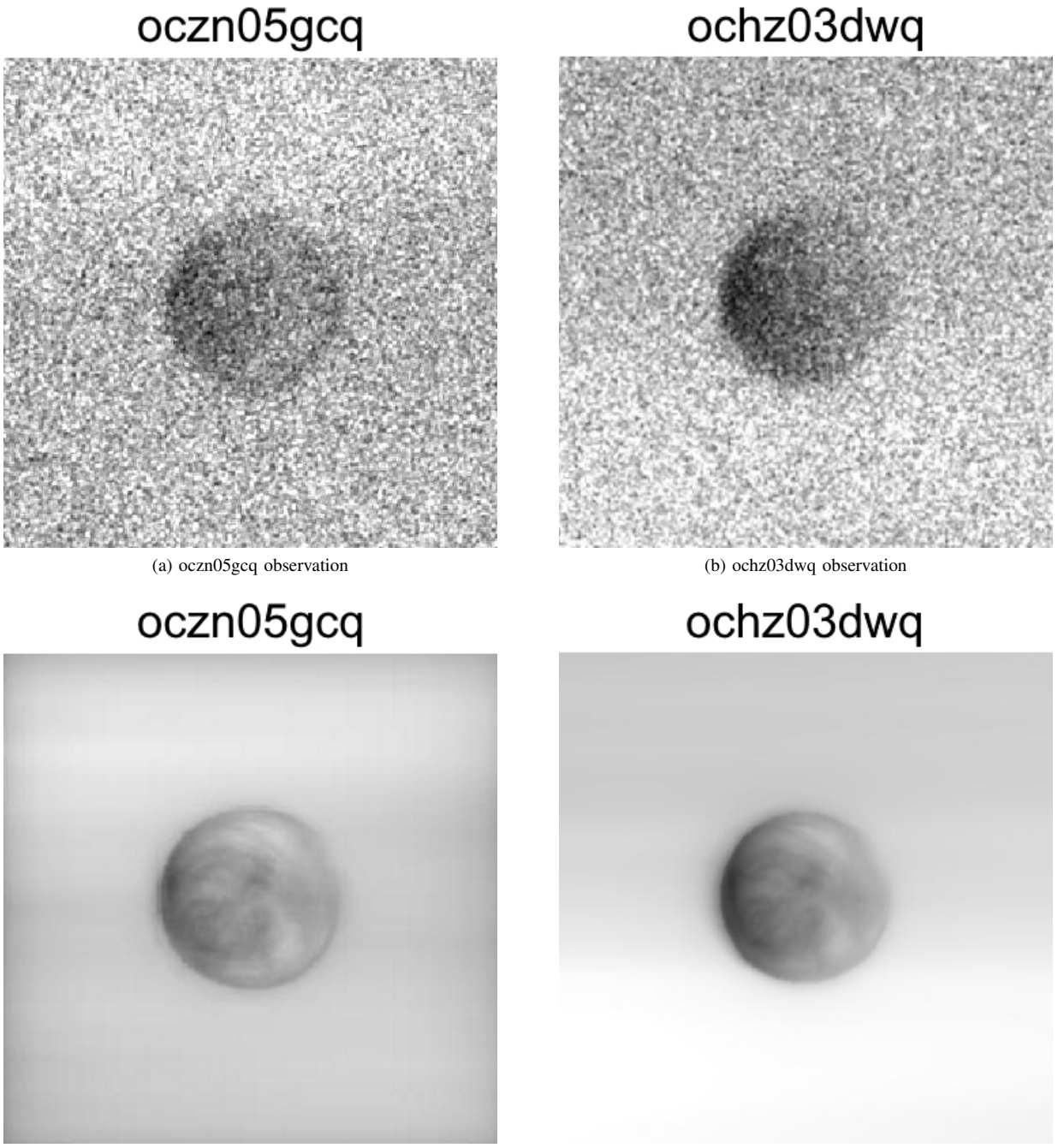
Several studies with the aim of finding water plumes on Jupiter's moon Europa has been conducted. The results of this study are highly relevant for two large future space missions, NASA's Europa Clipper Mission and ESA's JUpiter ICy Moon Explorer (JUICE). The Europa Clipper Mission will study the conditions for potential habitability at Europa. A spacecraft in orbit of Jupiter will make multiple flybys over the moon, each time it will alter the flight path to scan most of the moon. The timing and the trajectory of the different flybys will be controlled to capture data from the most interesting regions [2]. The results from this project will provide information that could be useful when determining where the flybys should take place in the future. ESA's JUICE mission will primarily focus on the moon Ganymede, but it will also explore Europa and Callisto to compare the different icy environments. Scientists believe that of all Jupiter's moons, Europa is most likely to harbor life. Important samples could be gathered by JUICE if water vapor plumes are found on Europa, these samples could then be studied with the hope of finding life [3].

This study is based on three previous reports, Sparks et al. 2016 [4], Sparks et al. 2017 [5] and Giono et al. 2020 [6], where plume existence has been discussed. In the first report by Sparks et al. 2016 three plume candidates were found. In Sparks et al. 2017 an additional plume candidate were found in a similar position as a previous candidate. The results from Sparks et al. 2016 were reproduced by Giono et al. 2020, where uncertainties in the method that has been used were found. However, Giono et al. 2020 claim that the plume candidates can be explained by statistical fluctuations and do not provide sufficient evidence for plumes on Europa.

In this project the algorithm developed by Giono et al. 2020 has been used to reanalyze the two observations, with observation ID oczn05gcp and ochz03dwq. These two observations are of interest since Sparks et al. 2017 found plume candidates in the same location. In addition, these two images of Europa have been combined and analyzed as a merged observation.

II. Observation

In this study, the two observations with ID oczn05gcq and ID ochz03dwq are of interest since Sparks et al. 2017 claim



(a) oczn05gcq observation

(b) ochz03dwq observation

(c) oczn05gcq model

(d) ochz03dwq model

Fig. 1. HST observations of oczn05gcq (left) and ochz03dwq (right) in the top row and their corresponding model in the bottom row.

that plumes are detected in the same location for these two. One of the images, ochz03dwq, was already analyzed in Giono et al. 2020. Our approach was to analyze the other one, oczn05gcq. Afterward, the two observations were merged and analyzed as a combined image.

The first analyzed observation of Europa, oczn05gcq, was obtained by HST 22-02-2016 using the F25QTZ longpass filter [5]. The second observation, ochz03dwq, was obtained 17-03-2014 using F25SRF2 longpass filter [4]. The filters are quite comparable and generate the same field of view and spatial resolution. They differ in that F25QTZ will on average day-side observing conditions cause about a 100 times lower sky brightness than F25SRF2, as a result of that F25QTZ has a better rejection of geocoronal emissions [7].

III. Method

HST data was downloaded and processed. A plume detection algorithm was applied to understand the statistical significance of the results. After the two observations oczn05gcp and ochz03dwq were analysed one by one, they were merged together and analysed as one further observation.

The plume detection algorithm is divided into two Matlab files. The first one reads the HST data and saves information about Europa's position and the sky brightness in the image. It was important to confirm that the correct filter was applied to the observation. The second Matlab file takes the information as input to produce a model that can be analysed. The two observations oczn05gcq and ochz03dwq and their corresponing models can be seen in Figure 1. Later on they were merged and analysed as an additional image. After the final models were composed and the position of the observations were confirmed the z-statistics was considered. A large negative outlier indicates a lower count of photons which in turn can be interpreted as a water plume absorbing light. The z-statistic is calculated as $z = (\langle I_{obs}\rangle_{5x5}/\langle I_0\rangle_{5x5} - 1)/\sigma$. I_{obs} refers to the photon count of the observation while the model photon count is I_0. Applied to the observation and the model was a 5 x 5 moving average filter (boxcar). $\langle I_{obs}\rangle_{5x5}$ and $\langle I_0\rangle_{5x5}$ denotes the resulting averaged observation and model respectively. The z-statistic is normalized by σ which corresponds to the uncertainty in the observation.

Adjustments in the algorithm were made throughout the process. Sky brightness, photons from unwanted sources, is examined in the code. In order to minimize the chance that a part of Jupiters surface was included in the caluculations of skybrightness the area that picked out pixels for the determination was decreased by a factor $\frac{1}{4}$. Figure 2 shows the observation image, sky brightness is calculated in the top left corner where the limb of Jupiter has subsided.

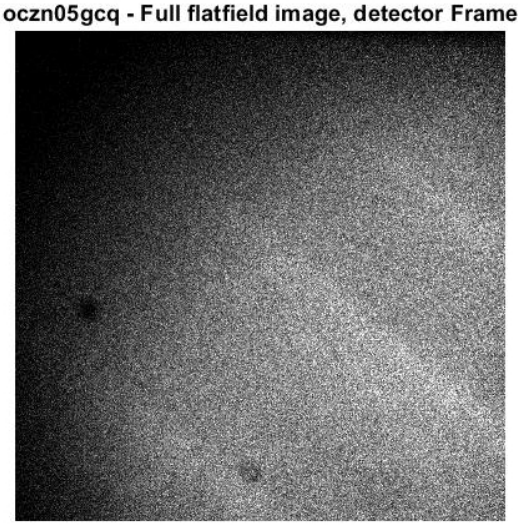


Fig. 2. The obesrvation oczn05gcq where Europa can be seen in transit of Jupiter. Europa is recognized as the dark anomaly at the lower center of the image.

Furthermore, the region that takes pixels to determine counts in the background were modified by decreasing it's size. Since a consequence of rotating the image when modelling the observation was that a section outside the image, that had zero photon counts was taken into account. This region outside the image can be seen in the top left corner of figure 3.

When modelling the observation, deviations were discovered at a long distance from Europa as a consequence of image convolution. Therefore, zoomed in images were used when generating the results in order to consider these aberrations.

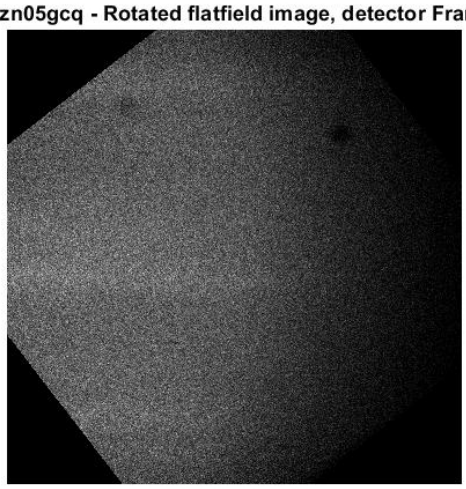


Fig. 3. Rotated image of the observation oczn05gcq.

TABLE I
SUMMARY OF RESULT FOR EACH OBSERVATION

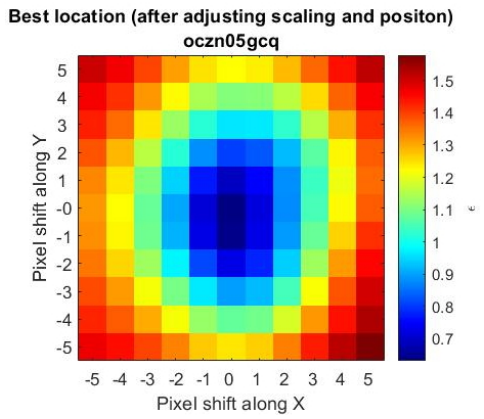
Observation ID	oczn05gcq	ochz03dwq	Merged observation
Date	22-02-2016	17-03-2014	—
Filter	F25QTZ	F25SRF2	—
Sky brightness	0.0280	0.4601	—
Largest negative outlier	3.8 σ	4.3 σ	5.1 σ
Largest positive outlier	-3.3 σ	-3.1 σ	-3.1 σ

Finally, the one-sided spectrum of the z-statistic was complemented by adding positive outliers in the same way as Giono 2020 since only the full statistical fluctuations allow a proper complete analysis. Giono 2020 argued that the positive side of the spectrum could not simply be ignored since their absence can be mistaken as plume absorption.

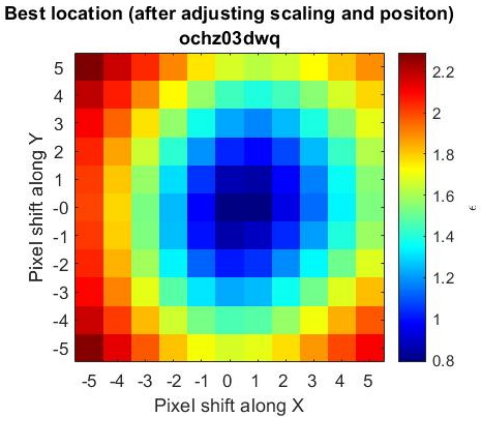
IV. RESULTS

Sky brightness is the average photon count from the background and is determined for both oczn05gcq and ochz03dwq. The oczn05gcq observation had an average sky brightness of 0.0280 counts and a standard deviation of 0.0922 counts. The ochz03dwq observation had an average sky brightness of 0.4601 counts and a standard deviation of 0.2273 counts. Different filters have been used while obtaining the oczn05gcq observation and ochz03dwq observation. Hence, the value of sky brightness of oczn05gcq was noticeable low. This will not affect the results since the sky brightness is subtracted from the observation before the z-statistics is determined.

To detect the location of Europa each observation was shifted vertical and horizontally relative to their corresponding model. Also, ϵ was determined for each shift combination. The ϵ metric quantifies the agreement between the model image and its corresponding observation. The limb angular profile is obtained by subtracting the observation from the model, ϵ is then calculated by taking the standard deviation of this angular profile. The best agreement of model and observation is then

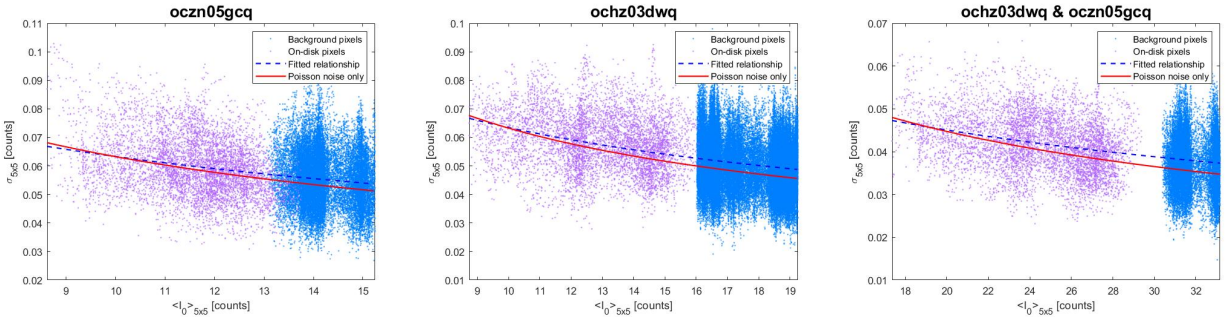


(a) Best location (after adjusting scaling and position) for the observation oczn05gcq.



(b) Best location (after adjusting scaling and position) for the observation ochz03dwq.

Fig. 4. Comparison of model and observation disk after adjustment of position and brightness scaling for oczn05gcq (left) and ochz03dwq (right).



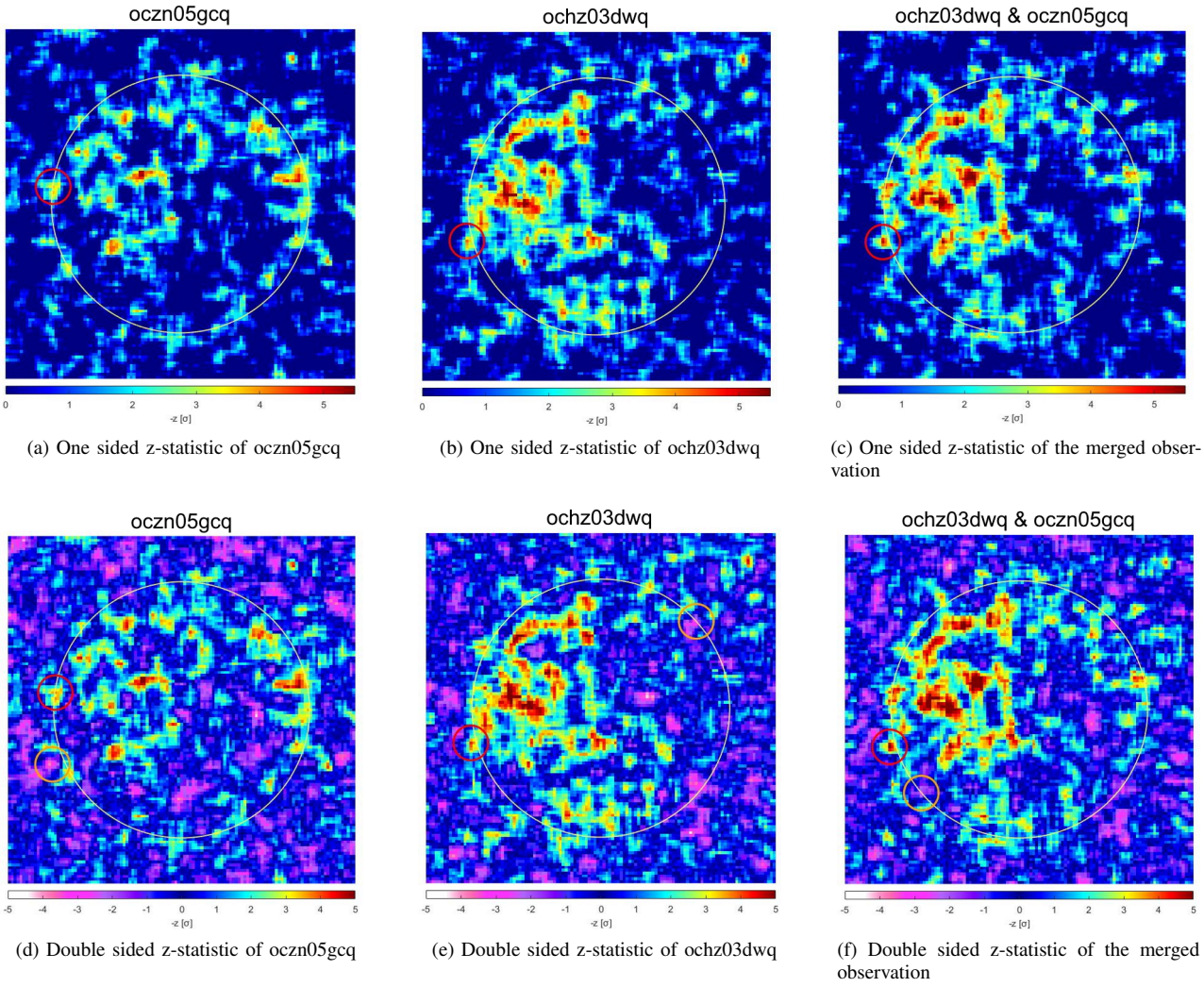
(a) oczn05gcp observation (b) ochz03dwq observation (c) ochz03dwq observation

Fig. 5. Plots of the relationship between model counts $\langle I_0 \rangle_{5x5}$ compared to the variation of the z-statistic inside each 5x5 averaged pixel for oczn05gcq (left), ochz03dwq (middle) and the merged image of oczn05gcq and ochz03dwq (left). The light blue dots are count rate of pixels from the background and the purple dots are the count rate of the pixels from the surface of Europa. The fitted relationship are shown by the blue dashed line which can be compared to the red line showing the relationship for only Poisson noise.

given by the case with the lowest value of ϵ. A larger value instead indicates a misalignment. As seen in 4a and Figure 4b the lowest value of ϵ was detected in ([0,0]). This corresponds to that the model and observation have the best agreement in these pixels. Therefore, the center and on-disk region (surface area of Europa) can be set.

From the given algorithm the z-statistic was obtained. The z-statistic is normally distributed and defines the difference between the model and observation. To normalize the distribution of z-statistic a relationship between σ which corresponds to the uncertainties in the model and the average of photon counts in a 5x5 pixel, I_{5x5}, was used. This fitted relationship for each observation is shown in Figure 5. Note that the differences in values between 5a, 5b and Figure 5b are a consequence of different mean count rates for each observation. The obtained z-statistic is shown in Figure 6 where the top row shows the one-sided z-statistic and the bottom row shows the double-sided z-statistic. The large yellow circle in each image indicates the position of Europa. Red circles mark the largest negative outlier in the limb region which provides a possibility of plume existence. The limb region refers to the region between Europa's surface and 5 pixels above the surface. The largest positive outlier around the limb, marked with yellow circles, is also of interest to discuss the significance of both positive and negative outliers. For observation oczn05gcq, in Figure 6d, the largest negative outlier has a value of 3.8 σ and the largest positive outlier has a value of -3.3 σ. Here the values refer to -z, this definition was used in Giono et al. 2020 and we choose to do the same. For the second observation ochz03dwq, in Figure 6e, the largest negative outlier has a value of 4.3 σ and the largest positive outlier has a value of -3.1 σ. When the observations were merged, in Figure 6f, the largest negative outlier has a value of 5.1 σ, and the largest positive outlier has a value of -3.1 σ.

The histograms of the z-statistic, in Figure 7, are assumed to be normally distributed around zero. The distribution of the z-statistic is examined for the disk (region inside Europa), the limb (region from the limb and 5 pixels away), and the background (outside the disk and limb region). For the case when the z-statistic is ideally normally distributed the value of σ_z should be a unity. However, as seen in Figure 7a, Figure 7b and Figure 7c all histograms have a $\sigma_z > 1$. The bigger deviation sigma has from a unity the more incorrect is the model of the observation. A major consequence of when the model does not represent the observation is that the z-statistic is improperly normalized. For all observations that have been



(a) One sided z-statistic of oczn05gcq

(b) One sided z-statistic of ochz03dwq

(c) One sided z-statistic of the merged observation

(d) Double sided z-statistic of oczn05gcq

(e) Double sided z-statistic of ochz03dwq

(f) Double sided z-statistic of the merged observation

Fig. 6. The top row displays the one-sided spectrum of the -z statistic for the observations oczn05gcq (left), ochz03dwq (middle) and the merged image of oczn05gcq, ochz03dwq (right). The bottom row instead display the double-sided spectrum of the -z statistic for oczn05gcq (left), ochz03dwq (middle) and the merged image of oczn05gcq, ochz03dwq (right).

analyzed the modeling of the background is better than the limb and on-disk region. Small deviations of σ from unity might be a consistent consequence of the noise present in the observations. The histogram of the z-statistic in the limb region for the merged observation in Figure 7c has the biggest σ which indicates the biggest mismatch between model and observation.

V. Discussion

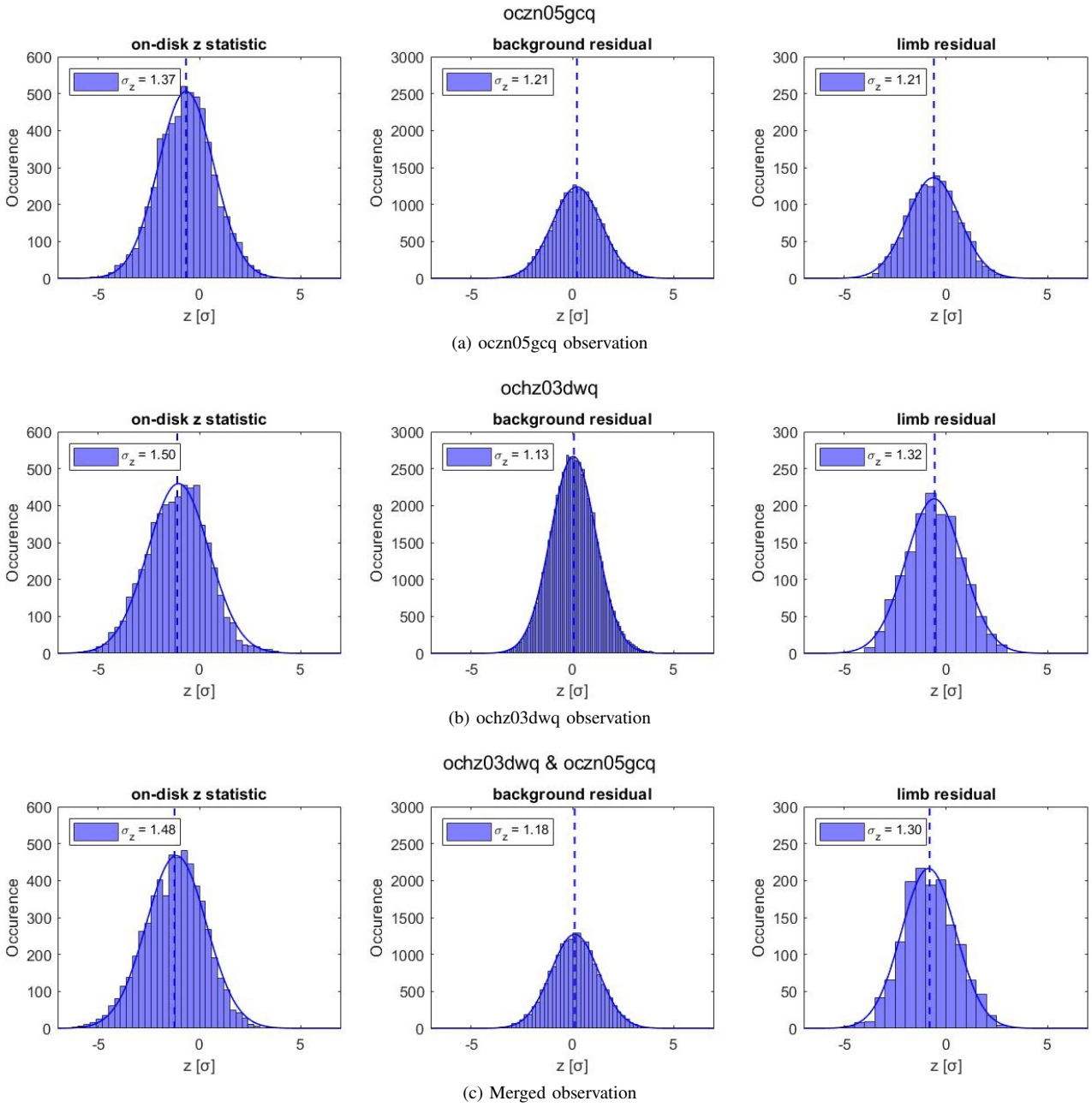
Assume that all obtained observations are independent. Also, assume that the limb is divided into 72 bins and that the possibility for events to occur is evenly distributed with a chance of 1/72 that an event will occur. From these assumptions, the probability can be calculated that an event will occur in the same bin twice. To get a chance over 50% that an event will occur twice, 11 images are required. In our case, we have 20 obtained images from the Hubble telescope. This corresponds to a chance of 94.58% that an event will randomly occur twice in the same bin.

The two observations chosen for this analysis were not randomly selected. They were selected because they have an outlier in the same bin according to Sparks et al. 2017. Hence, this is no longer a random experiment.

According to Giono et al. 2020, an outlier with a significance of 3.2-sigma is expected in every image, which is based on the number of pixels and assuming Gaussian noise.

The first analyzed observation, oczn05gcq, provides an outlier with a significance of 3.8-sigma. Which is a larger significance than 3.2 that can be expected in every image. The largest positive outlier has a value of -3.3 sigma which corresponds to the expected value. Therefore, the negative outlier can be considered as a plume candidate. However, the outlier did not occur in the same location as for Sparks et al. 2017. The plume candidate we found in this observation might rather be an effect of statistical fluctuations as a consequence of uncertainties between the model and observation.

The second analyzed observation, ochz03dwq, provides the largest negative outlier with a significance of 4.3-sigma, which is a noticeable larger significance than what can be expected.



(a) oczn05gcq observation

(b) ochz03dwq observation

(c) Merged observation

Fig. 7. Histograms showing the z-statistic for oczn05gcq (top row), ochz03dwq (middle row) and the merged image of oczn05gcq and ochz03dwq (bottom row). The left column shows results from the on-disk region, the middle column for the background region and the right column for the region on the limb. The dashed lines show the average value for the z-statistics while the blue curved line i a Gaussian fitting.

The largest positive outlier had a value of -3.1-sigma which corresponds to the expected value. Also, the plume candidate is detected in the same location as expected. The possible plume existence in this image can not be excluded even if statistical fluctuations exist.

The merged observation can not be judged in the same way as the individual observations. It is not a random experiment anymore since the two merged observations were selected because they provide plumes in the same location according to Sparks et al. 2017. Therefore, we expect an outlier with a large significance in a specific location. The largest negative outlier in this experiment had a value of 5.1-sigma and was detected in the same location as expected. However, the high probability of 94.58% to redetect a plume candidate in the same bin together with the fact that these observations were selected because they had a plume candidate in the same location shows that this is not a random result.

VI. Conclusion

The possible existence of plumes around the limb of Europa would be important knowledge for upcoming space missions such as the Europa Clipper Mission and JUICE since the environment will affect scientific instruments during the exploration. Furthermore, there is a lot of interest in searching for life in samples from the ocean under Europa's icy crust. Letting space probes fly through a water plume to collect

samples could be the most promising way to go. That's why finding these hypothetical plumes is important. In recent years several reports have shown evidence for water plumes around the limb. This report has been an attempt to independently verify some of these claims and contribute to an eventual finding of water plumes on Europa.

The two analyzed observations were identified by Sparks et al. 2017 to have plume candidates in the same location. The analysis of the first observation, oczn05gwq, did not indicate plume existence in the previously found location. For the second observation, ochz03dwq, a 4.3 σ outlier occurred in the same bin as Sparks 2017 expected. The possibility of plume existence is not completely excluded. Although, misalignment between observation and model generates a distorted statistic. Noticeable is that the largest negative outlier in the analysis of the second observation, ochzn03dwq, increased when it was merged with the first observation oczn05gcq. The plume candidate in the merged observation has a statistical significance of 5.1 σ, which is not a random result since the combined observations were chosen because they have provided plumes in Sparks et al. 2017.

There are some uncertainties in the conclusion of plume existence. The distribution of z-statistic has a σ bigger than unity. This indicates a misalignment between the model and observation. These local mismatches lead to improperly normalized z-statistic and therefore distorted statistics. Also, the two combined observations were non-random observations. They were chosen due to that Sparks et al. 2017 pointed out that these two are similar observations with possible plume existence in the same location. Therefore there are difficulties to state that this is a random result that favors the theory of plume existence.

In the future, one way to strengthen the theory of plume existence could be to combine observations randomly. Also, develop a safer method where the chance of statistical fluctuations is minimized. In conclusion, statistical fluctuations need to be excluded to provide the possible plume existence.

ACKNOWLEDGMENT

The authors would like to thank the supervisor Lorenz Roth for the help and guiding through the project.

REFERENCES

[1] C. J. Hansen, L. Esposito, A. Stewart, J. Colwell, A. Hendrix, W. Pryor, D. Shemansky, and R. West, "Enceladus' Water Vapor Plume," *Science*, vol. 311, no. 5766, pp. 1422–1425, Mar 2006.

[2] J. R. Thompson. (2021, Apr) About the Mission. NASA, Washington, D.C, USA. [Online]. Available: https://europa.nasa.gov/mission/about/

[3] O. Grasseta, M. Dougherty, A. Coustenis, E. Bunce, C. Erd, D. Titov, M. Blanc, A. Coates, P. Drossart, L. Fletcher, H. Hussmanni, R. Jaumann, N. Krupp, J. Lebreton, O. Prieto Ballesteros, P. Tortora, F. Tosi, and T. Van Hoolst, "JUpiter ICy moons Explorer (JUICE): An ESA mission to orbit Ganymede and to characterise the Jupiter system," *Planetary and Space Science*, vol. 78, pp. 1–21, Feb 2021.

[4] W. B. Sparks, K. P. Hand, M. A. McGrath, E. Bergeron, M. Cracraft, and S. E. Deustua, "PROBING FOR EVIDENCE OF PLUMES ON EUROPA WITH HST/STIS," *The Astrophysical Journal*, vol. 829:121 (21pp), Oct 2016.

[5] W. B. Sparks, B. E. Schmidt, M. A. McGrath, K. P. Hand, J. R. Spencer, M. Cracraft, and S. E. Deustua, "Active Cryovolcanism on Europa?" *The Astrophysical Journal Letters*, vol. 839:L18 (5pp), Apr 2017.

[6] G. Gionno, L. Roth, N. Ivchenko, J. Saur, K. Retherford, S. Schlegel, M. Ackland, and D. Strobel, "An Analysis of the Statistics and Systematics of Limb Anomaly Detections in HST/STIS Transit Images of Europa," *The Astronomical Journal*, vol. 159:155 (14pp), Apr 2020.

[7] David J. Shayler and David M. Harland. (2021, Feb) The Hubble Space Telescope User Documentation. [Online]. Available: https://hst-docs.stsci.edu/stisihb/chapter-14-imaging-reference-material/14-5-fuv-mama/f25srf2-fuv-mama-longpass

CONTEXT I
SUBORBITAL FREE FLYER FOR NEAR-EARTH SPACE RESEARCH

POPULAR DESCRIPTION

Space research - The return of the experimental unit

Is it a bird? Is it a plane? No, it is a flying science lab! 300 seconds ago, it was launched to the brink of space on a sounding rocket from Esrange Space Center near Kiruna at three times the speed of a bullet. It is now plunging towards the Earth. Surely there must be a parachute stored inside, allowing for a soft landing. Suddenly, wings and a tail are deployed. The autonomous flight control system activates… The science lab is heading home!

In cutting-edge space research, sounding rockets are commonly used to send out miniature science labs in the form of experimental units, intended to perform measurements in near-earth space. When using parachutes for recovery, these units will be landing at random locations scattered over an enormous area. This often comes with unwanted complications such as expensive search expeditions with helicopters. Many units are even lost forever and never found in the endless wilderness of northern Sweden. The limited hours of daylight further restrict the search expedition and in other areas of the world, the experiments could be lost at sea.

The solution to this problem is making the experimental unit return to the launch site autonomously. Named after the aboriginal hunting tool, the BOOMERANG team at KTH Royal Institute of Technology has the purpose of completing this exciting objective. Designing the autonomous glider solution requires a multidisciplinary team of engineering students. Students with a clear vision: “Rocket payloads used to go BOOM and now they go BOOMERANG!”

SUMMARY OF PROJECT RESULTS

Society today relies heavily on the use of space systems, which provide the basis for human communication and navigation. To know the conditions in which these systems operate, and the effects of space phenomena on the global atmospheric system, one must understand the near-earth space region. Due to satellite orbits not being stable below an altitude of 300 km, the lower parts of the ionosphere are not as extensively studied as space higher up.

This has made studies of the region between 50 km and 200 km territory of the sounding rockets - suborbital probes that return to earth after a relatively short trajectory. Recently, sounding rockets carrying payloads in the form of experiments or measuring devices have gained interest since they allow studying near-earth space. The German Aerospace Center (DLR), the Swedish National Space Agency (SNSA), and the European Space Agency (ESA) collaborate annually to provide sounding rockets for students to carry out scientific and technological experiments via the REXUS program, which KTH Royal Institute of Technology has been active in.

The payloads are typically realized by multiple units carried by the same rocket that may be ejected at a certain altitude. Once ejected, the now Free-Falling Unit (FFU) will take measurements and store data of interest as it plunges toward the earth. Because of the sounding rocket's trajectory, the FFU will land far away from the launch site and need to be retrieved by helicopter. Retrieving the FFU by helicopter works in practice, but is both costly and time-consuming. In the worst-case scenario, the units might be lost forever along with their experimental data.

The desire to have the FFU autonomously fly back to the launch site instead birthed four BSc projects focused on different aspects of the FFUs recovery unit (RU). The project groups within ”Context I” are now tasked with contributing to this objective by proposing solutions for how the FFU should glide (**I1**), deploy its wings (**I2**), trajectory and attitude control (**I3**), and the electrical power system (**I4**). Additionally, these projects aim to be part of future REXUS programs.

Project group I1 is responsible for the overall aerodynamics of the glider. This entails the design and decision of airfoil, wing, tail, control surfaces, and mass placement. Considering the dimensions and flight requirements of the FFU, a prototype glider

solution given the name ICEBERG is designed and modeled in the aerodynamics simulation software XFLR5. Similar to an ordinary aircraft, this consists of a wing and tail for the FFU, the latter chosen as a non-conventional inverted V-tail to avoid the turbulent wake from the FFU fuselage. Furthermore, the wing has a minor dihedral for lateral stability and the glider has a center of gravity close to the front for longitudinal stability. These design features, amongst others, are critical for a successful flight of the future glider solution after ejection from the REXUS rocket.

Just like any iterative aircraft design process, flight testing of ICEBERG in software and real-life will result in the design of future prototypes. Some potential improvements include more detailed aerodynamic shaping of the FFU without risking its integrity, the addition of winglets for reduced induced drag, and larger control surfaces for better controllability.

Project group I2 focuses on developing the deployment mechanism of the wings for the FFU and analyzing the structural stability of the system. The goal is to create a design capable of unfolding to a larger specific shape that is both aerodynamic and structurally robust, to handle the mechanical stresses induced by launch vibrations and the pressure and inertial loads during flight. The finalized design is composed of a Scissor Structural Mechanism (SSM) to achieve the desired length and airfoil ribs to achieve the desired shape. Additionally, a compression spring and a thermal cutter will trigger the deployment system.

The analysis shows that the deployment system will fit inside the RU if the wings are placed at the center. Furthermore, it shows that the shear stress will not exceed the safety parameters for appropriate material selections. Currently, the deployment system only focuses on the wings. Future work includes developing a deployment mechanism for a tail, which is crucial for the stability and controllability of the FFU.

Project group I3's target is to design, develop and implement the autonomous flight control system for the FFU. The purpose of the flight control system is to make sure the FFU maintains the proper attitude and heading during its descent, which is realized by two independent control systems for the longitudinal and lateral aerodynamics respectively. The control systems utilize a total of three Proportional Integrating (PI) controllers as well as a Proportional (P) controller that is structured such that there is an inner control loop for stability and an outer control loop for direction for both longitudinal and lateral modes.

The controllers, sensor filtering, and the logic necessary to steer the FFU towards a given location are implemented in a SIMULINK simulation environment using data from XFLR5 (provided by project group I1) for the physics engine. The SIMULINK open-loop simulation provides expected results, matching the poles and steady-state conditions predicted by XFLR5. The simulation results also show that it is feasible to use the proposed control system structure as a satisfactory autopilot, given that the glider design is relatively stable or near-stable.

In future projects, one could further investigate and compare the choice of control, improve filtering of sensor data and optimize the final C-code on the hardware. Improvements in these areas will lead to faster and more accurate responses reducing the risk of unstable flight.

Project group I4 is responsible for providing the required power for the FFU to perform its autonomous flight. This is done by designing, producing, and testing a prototype of a motherboard with integrated motor drivers and a flight control system. This motherboard is designed to control the position of two brushed DC motors, provide necessary control surface feedback and deliver power to the necessary systems such as GPS and Datahub with integrated microcontroller and Field Programmable Gate Array (FPGA). This will become the platform for project group I3 to run their autonomous flight control system on.

The output torque from the two motors is adjusted for the torque needed to turn the control surfaces designed by project group I1. Furthermore, a simple robust signal triggered deployment mechanism for project group I2 is implemented and all this is packaged in a small FFU compatible format. Further development could be done in the possibility of implementing a different kind of battery solution not limited to 1 cell SAFT brand batteries.

Future plans include merging all I-context projects for a final version of the Suborbital Free Flyer. After convincing test results, the final glider solution will be launched and flown with a REXUS sounding rocket in 2023. Some of the bachelor's thesis students hope to continue within the project even after the thesis.

IMPACT ON SOCIETY AND ENVIRONMENT

In contrast to other modern research topics such as AI and genetic research, space research with sounding rockets, and the experiments within the context I do not create any direct impact on society or lead to substantial ethical dilemmas. This is mainly because the conducted research in REXUS seeks to bring a greater understanding of how this region of the atmosphere

works and behaves. In other words, the nature of the conducted research is that of data gathering, analysis, and understanding as opposed to testing new theories and hypotheses. However, the work done in the context is still subject to some environmental and social effects and two main ethical dilemmas worth discussing.

First of all, the cost of space research projects such as this one is normally funded by the government. This money could be used for socio-economic means, for example, education, health care, expanding infrastructure, renewable energy sources, and so on. Also, one could argue that other research fields such as the treatment of cancer are more important than space plasma physics. On the other hand, scientific research is fundamental for technological development, which in the future can enhance human conditions. For example, understanding plasma in near-earth space can potentially lead to economic savings if, or even when, space travel becomes more accessible. Additionally, fusion reactors require plasma to produce power, which means studying plasma in space could help advance that technology.

Since the technology being developed in this project further enables the study of plasma in near-earth space, it is also conceivable that the REXUS experiments yield useful results regarding solar wind and storms. Such results could prove very impactful on society since solar storms can cause disruptions in electronic communications. Having functioning electronic communications is so important that the Swedish Civil Contingencies Agency listed solar storms as one of 27 national risks in the Swedish National Risk Assessment 2012. For the function of society as we know, it could then prove vital to have performed such experiments, for the appropriate proactive and reactive measures to be taken. Consequently, the project is deemed meaningful both economically and socially in the long run.

Secondly, the rocket engines used in REXUS are surplus military gears, which raises another ethical dilemma. By sourcing expired rocket engines from the military, one could argue that as this indirectly supports the military industry, this is an unethical deed. The military might have not had an altruistic use of the materials and unfortunately, there exist questionable political and ethical convictions within the industry that should not be looked over. However, one of the main advantages is the fact that unused materials and parts are being used, which reduces the environmental impact compared to if the equipment were to be scrapped. Additionally, by reusing already functional and tested technology, there is no need for developing a new rocket engine. This improves safety and reduces the risk for faulty rockets exploding which could lead to less debris.

Even though the environmental and societal impacts of this research type is not directly obvious, in light of what has been discussed in this section, the project members of context I would still argue that the benefits of exploring the near-earth space outweigh the consequences and that further enabling such research through the use of autonomous experiment units only lessens the negative impacts.

High Altitude Glider Solution for Returning From Space

Koray Amico Kulbay and Jakob Nylöf

***Abstract*—Space exploration drives the human expansion in the universe. Succeeding in this challenge demands familiarity of the near earth space environment, achieved through sounding rocket experiments that often are lost upon return from space. A future proof solution is needed and this study aims to investigate the aerodynamics of a modular self returning glider attachment.**

To aid conceptual design, simulations were first performed using potential theory in the software XFLR5. The resulting design was then analysed further using Computational Fluid Dynamics (CFD) in Simscale after which a glider prototype was built and tested.

The study shows that while it is possible to fulfill the project requirements when only modelling the wing surfaces, the glider fuselage contributes to a destructive drag and pitching moment. Consequently, future prototypes demand increasing the lift or reducing the drag, as well as ensuring longitudinal stability. More resources need to be invested into further CFD modelling and prototype testing.

***Sammanfattning*—Utforskning av rymden driver den mänskliga expansionen ut i universum. För att lyckas med det krävs dock kunskap om rymden närmast oss, vilket uppnås genom experiment i sondraketer som ofta förloras vid återkomst. En framtidssäker metod behövs och därför undersöks aerodynamiken av en modulär och självåtervändande glidarlösning.**

För att underlätta genomförandet av den konceptuella designen så gjordes först simuleringar i XFLR5 med potentialteori. Den resulterande glidaren analyserades sedan vidare i flödesmekaniska beräkningsprogram (CFD), vartefter en prototyp byggdes och testades i verkligheten.

Studien visar att det är möjligt att uppfylla projektkraven genom att modellera vingarna, men glidarens flygkropp bidrar emellertid till ett destruktivt luftmotstånd och longitudinellt vridmoment. Därför måste framtida prototyper designas för att uppnå större lyftkraft, minska flygkroppens dragkraft och samtidigt uppnå longitudinell stabilitet. Mer resurser måste läggas på djupare CFD-modellering och prototyptestning.

***Index Terms*—Aerodynamics, aeronautics, gliding flight, near-earth space research, sounding rocket, KTH, REXUS, XFLR5, Simscale, CFD.**

Supervisors: Mykola Ivchenko, Raffaello Mariani

TRITA number: TRITA-EECS-EX-2021:165

I. Introduction

The REXUS program is a possibility for student experiments to access the upper atmosphere through an agreement between the German Aerospace Center (DLR) and the Swedish National Space Agency (SNSA) in collaboration with the European Space Agency (ESA) [1]. Student teams from KTH Royal Institute of Technology have been a part of this program for several years. Commonly the Free Falling Units (FFUs) containing the scientific instruments from KTH are retrieved through a landing parachute, which results in both a time consuming and costly retrieval. An optimal solution would be to have the FFU eject from the REXUS Rocket Mounted Unit (RMU), perform any arbitrary experiment, fall towards the earth, deploy a glider solution from the FFU and then autonomously return to the launch site. Such a future vision is the core purpose of this very project and the latest student experiment team at KTH called BOOMERANG (BOOM-deploying Experiment with Return-to-launch-site Automated Non-propelled Glider). The team aims for a launch in 2023 with the mission patch seen in figure 1.

Fig. 1. Mission patch of the KTH student team BOOMERANG, designed by team member Márton Galbács.

This study aims to design a suitable glider solution and analyse its aerodynamic properties. This essentially requires modelling lift, drag and moment contributions of wings, tail and the cylindrical FFU which will come to serve as a fuselage. Furthermore, I1 is in close collaboration with the BOOMERANG team and three other bachelor thesis groups: I2 (deployable wing structure), I3 (glider control system) and I4 (glider electrical power system). The project is supervised by associate professor Mykola Ivchenko.

II. Theory: Fundamental Aerodynamics

A. Fundamentals of flight

Looking to the sky one can easily spot an ”iron bird” soaring past, seemingly without effort. The physics behind an airplane’s ability to fly boils down to the resultant shear and pressure forces acting upon it [2]. These interactions can be separated into forces and moments that eventually translates into flight.

The forces acting perpendicular to the the airflow, represented by the free stream velocity vector V_∞, are summarized

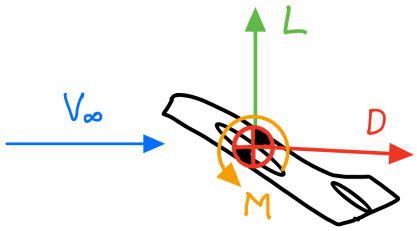


Fig. 2. Fundamental sources of flight in relation to the free flow here illustrated for 2-dimensional flight. L is the force of lift, D the force of drag and M the pitching moment acting on the glider. V_∞ is the velocity of the air stream known as the free stream velocity.

as *lift* L and the the forces parallel to it as *drag* D [3]. Interestingly these forces result in a moment acting on the glider [3]. It is also through moments and uneven force distributions airplanes are controlled, which is further discussed in section V-B. In figure 2 these three fundamental sources of flight can be seen in relation to each other. This opens up the world of aerodynamics.

B. Usage of coefficients

Aerodynamic coefficients are a way to represent the forces and moments as dimensionless quantities. The usage of coefficients has proven very efficient in aerodynamics as the forces and moments depend on many variables [3]. The coefficients also facilitate comparison of plane designs. For lift, drag and pitching moment acting on a glider they are listed below, but are defined analogously for other forces and moments.

1) Coefficient of lift:

$$C_L = \frac{L}{q_\infty S} \tag{1}$$

where $q_\infty = \frac{1}{2}\rho_\infty V_\infty^2$ is called the dynamic pressure, S is the wing area and ρ_∞ is the density of the medium (in this case air) [3].

2) Coefficient of drag:

$$C_D = \frac{D}{q_\infty S} = C_{D,0} + C_{D,i} = C_{D,0} + \frac{C_L^2}{\pi e AR} \tag{2}$$

where $e \leq 1$ is the Oswald efficiency number and $AR = b^2/S$ the aspect ratio with b as the wing span [4] [3]. $C_{D,0}$ is the drag at zero lift and $C_{D,i}$ is the induced drag.

Equation 2 reveals an interesting insight into the drag of a plane. The first term is the drag of a wing of infinite wing span, meanwhile the second term is the "punishment" of a finite wing. This increase in drag is due to the loss of lift when airflow escapes over the finite wing tips to the top of the airfoil instead of under it [4].

3) Moment coefficients: Figure 3 from [5] illustrates the moments that act on a glider in different directions. The directions are the axes of a glider-fixed coordinate system known as the body axis system. The moment M causing the glider to pitch up and down is called pitching moment as is represented by the following coefficient

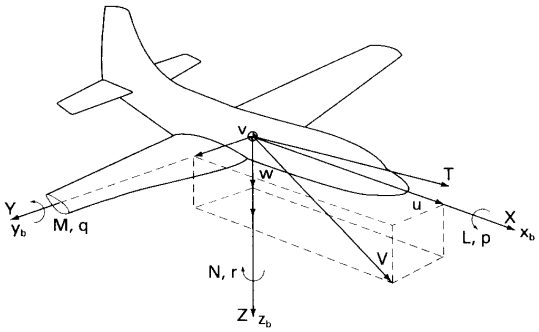


Fig. 3. Diagram from [5] showing the pitching, yawing and rolling moments M, N and $L = L'$ (not lift) acting on an airplane in the body axis system $x_b y_b z_b$. The vector quantities for angular velocity $\omega = (p, q, r)$, velocity $\mathbf{V} = (u, v, w)$ and aerodynamic force $\mathbf{F} = (X, Y, Z)$ in the body axis system are also illustrated. Note that the thrust vector $\mathbf{T}$ is not present when considering a glider.

$$C_m = \frac{M}{q_\infty S c} \tag{3}$$

where c is the cross sectional length of the wing known as the hord. Analogously, the moment N causing a left and right turn is called yawing moment, and is represented by the coefficient C_n, while the moment L' (prime to not confuse with lift) causing the glider to rotate about its central axis is called rolling moment and its coefficient is denoted by C_l [3] [5].

C. Sources of drag

Investigation of a wing is initially based on the analysis of the airfoil, which is the geometric shape of the wing cross section.

Considering the airfoil drag there are three sources: wave drag (can be omitted when considering subsonic flight) [3], skin friction drag and pressure drag where the last two commonly are referred to as profile drag [3].
The last source of drag of a finite wing is the induced drag as discussed in section II-B2.

D. How lift is attained

Ever since mankind harnessed the power of flight alternative explanations of lift have been employed by different scientists and engineers. The following is the most fundamental reasoning, according to the authors and many others, and is deduced in [3]:

1) **Squashed airflow and mass continuity:**
   As seen in figure 4 the cross sectional area (A) is smaller on top of the airfoil compared to under it (due to its special shape and/or angle of attack, α. A positive angle of attack is when the front of the airfoil forms an angle with the free flow direction of travel). Mass continuity $\rho AV = constant \implies$ velocity (V) increase when the cross sectional area decreases both above and below the airfoil. However, $A_A < A_B \iff V_A > V_B$.

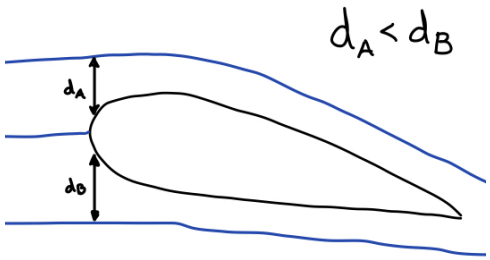


Fig. 4. Squashed airflow over airfoil.

2) **Bernoulli effect:**
Due to the Bernoulli equation for incompressible flow and the Euler equations for compressible flow, a larger flow speed and thus lower pressure is experienced on top of the airfoil compared to under it.
3) **Pressure distribution:**
A pressure gradient is created and a positive lift force acts upwards on the airfoil.

Et voilà, lift is attained!

E. Effect of Mach number

Mach number is a way to relate the aircraft speed to the speed of sound. This is of interest due to the strange [3] aerodynamical effects that occur when an aircraft approaches the speed of sound. That specific state of speed is called Mach 1, with Mach number defined as follows:

$$M = \frac{V_\infty}{a} \tag{4}$$

where M is the Mach number and $a(h)$ is the speed of sound [3]. h indicates the dependence of altitude to determine the speed of sound [3].

In order to avoid the effects mentioned above, larger drag [3] and using compressible flow [3] many applications limit their aircraft to $M < 0.3$, which has been the case of this very project.

F. Glide performance

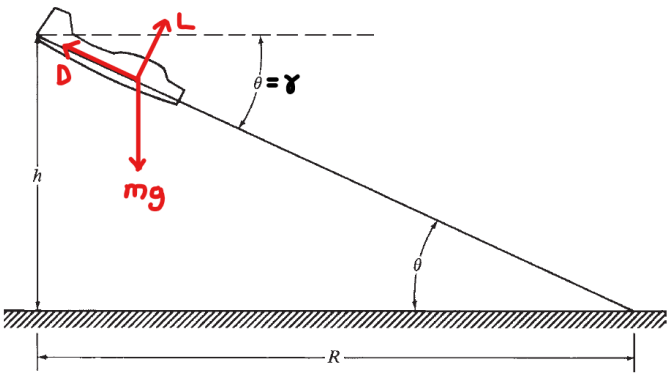


Fig. 5. A schematic image of equilibrium gliding flight over range R and height h retrieved from [3]. Note that the glide angle $\gamma = \theta$. The forces of motion acting on the glider in equilibrium [3] has been added to the image, illustrating the derived relations.

The flight equations of motion for gliding flight are derived from Newton's second law, where the only forces acting on the glider are lift, drag and gravity. When considering steady level (see section III-C) non-accelerated flight as in figure 5 the force equilibrium equations in the wind axis system become [3] [6] [2]

$$D = mg \sin\gamma \tag{5}$$
$$L = mg \cos\gamma \tag{6}$$

where γ is the positive angle between the glide path and the horizontal, m is the mass of the glider g the gravitational acceleration. For the FFU (the cylindrical glider body) returning to Esrange, interest lies in maximising the range R i.e the travelled horizontal distance of the glider, which is equivalent to maximising the glide ratio $G = L/D = C_L/C_D$ [2] as is shown later in this section. Division of equation 5 & equation 6 yields an expression for the glide ratio as the inverse of the glide angle due to small angle approximation [2]

$$\frac{L}{D} = \frac{1}{\tan\gamma} \approx \frac{1}{\gamma}. \tag{7}$$

Furthermore, glide ratio can be expressed as the quotient of R and travelled vertical distance h [6] [3]. However for varying glide ratio, this instead becomes a derivative

$$G = \frac{L}{D} = \frac{dR}{dh}. \tag{8}$$

Additionally due to small angle approximation $L = mg\cos\gamma \approx mg$. Rewriting equation 1 yields an expression for the speed of the glider [2] [6]

$$V = \sqrt{\frac{2}{\rho C_L}\frac{L}{S}} = \sqrt{\frac{2}{\rho C_L}\frac{mg}{S}}. \tag{9}$$

Using $\sin\gamma \approx \gamma \approx C_D/C_L$ the projected vertical speed or sink rate becomes

$$\dot{h} = -V\sin\gamma = \sqrt{\frac{2mg}{\rho S}}\frac{C_D}{C_L^{3/2}} \tag{10}$$

which means the time of flight TOF is readily derived [6]

$$TOF = \frac{\Delta h}{\dot{h}}. \tag{11}$$

However, for gliding flight in the atmosphere at heights between 0 and 10 km the density changes substantially [2]. This causes the above quantities, and specifically glide angle for maximum range to slightly change during descent [6]. In other words, the glide angle and thus glide ratio is essentially a function of altitude $L/D = G(h)$. Integrating equation 8 yields

$$R = \int_{h_1}^{h_2} G(h)dh \tag{12}$$

which implies that R is maximized by flying at the glide angle corresponding to maximum glide ratio for all heights h. This will be important for control, which will aim to trim the glider's pitch angle to coincide with the glide angle corresponding to maximum glide ratio [6].

III. Theory: Stability

If creating a functional aircraft was just as easy as attaching two wings to an object a lot of aerospace engineers would be out of jobs. An aircraft needs to be *stable* to function as intended. Stability can be divided into two main subgroups: *static* and *dynamic* stability.

A. Static stability

"The initial tendency of an object, when disturbed from equilibrium, to return to that condition."
- Raffaello Mariani [7].

There exists three types of static stability that refer to the planes' orientation in a three dimensional space.

1) ***Longitudinal - Pitch****:* Longitudinal stability is obtained when a plane counteracts a nose up or down movement, called pitch. This translates into the following relations:

$$\frac{\partial C_m}{\partial \alpha} < 0 \tag{13}$$

$$C_m(\alpha_{L=0}) > 0 \tag{14}$$

where C_m is the pitching moment coefficient, α is the angle of attack and $\alpha_{L=0}$ is the angle of attack where the lift is zero. $C_m(\alpha_{L=0})$ is referred to as the zero-lift moment [7].

The relations above are primarily satisfied and altered with the choice of main wing and horizontal tail.

2) ***Lateral - Roll*** *:* Lateral stability is obtained when a plane counteracts a rotating movement, called roll. This translates into the following relation:

$$\frac{\partial C_l}{\partial \beta} < 0 \tag{15}$$

where C_l is the roll moment coefficient, β is the sideslip angle. The sideslip angle is the angle between the aircraft's direction of travel and direction of the nose.

This relation is mainly satisfied with dihedral, which is a specific design choice of the wing attachment further discussed in section V-A.

3) ***Directional - Yaw****:* Directional stability is obtained when a plane counteracts a right or left turning movement, called yaw. This translates into the following relation:

$$\frac{\partial C_n}{\partial \beta} > 0 \tag{16}$$

where C_n is the yawing moment coefficient.

This relation is mainly satisfied with the vertical tail.

In aerodynamics it is common practice to use chain indexing to write different variations of the coefficients. Coefficient derivatives are written by chain indexing of their respective variables e.g. $C_{m_\alpha} := \frac{\partial C_m}{\partial \alpha}$ [7].

B. Dynamic stability

"The response of the system over time."
- Raffaello Mariani [7].

Studying the time dependence of aircraft motion can be performed in different ways [5], but the authors have used the root locus technique in this project. Roots can be attained through the characteristic equations from the aircraft's equations of motion. They reveal the oscillatory behaviour, convergence speed and magnitude of aircraft modes. The modes critical for dynamic stability are listed below [5].

1) Longitudinal modes:

- Phugoid: Oscillatory change in altitude.
- Short-period: Oscillatory change in angle of attack.

2) Lateral modes:

- Dutch roll: Oscillatory motion of coupled roll and yaw.
- Spiral: Non oscillatory spiral movement that can result in a steep dive.
- Directional: Non oscillatory movement of increasing sideslip that results in a curved flight path.

C. Steady flight

An aircraft's native flight state is called steady flight, which occurs when *all* moment coefficients mentioned in section III-A and forces are in equilibrium.

IV. Theory: Maneuverability

Just like any aircraft, gliders can be maneuvered during flight using various control surfaces on the wings and tail made to adjust roll, pitch and yaw [5]. They allow the pilot, or in this case an autopilot, to control the glider's direction of travel.

In this project the aim has been to only control pitch and yaw. The following sections describe how this is done and how to estimate the resulting moments on the control surface hinge axles.

A. Controlling pitch

The elevator is the control surface on the horizontal tail which controls the pitch. Let the angle of deflection of the elevator in reference to the chord line of the airfoil be δ_e which is defined positive for deflection downwards. Deflecting the elevator results in linear terms adding to the pitching moment and lift coefficient of the glider. The change in pitching moment is the main focus. The pitching moment after addition of a deflected elevator flap is

$$C_m = C_m(\alpha_{L=0}) + C_{m_\alpha}\alpha + \Delta C_m \tag{17}$$

$$\Delta C_m \propto -V_H \tau \delta_e \tag{18}$$

where V_H is the horizontal tail volume (defined in V-B) and τ the flap effectiveness which increases with the fraction of the elevator area over the tail area S_e/S_t. This formula shows that changing the deflection angle δ_e shifts the moment coefficient curve along the C_m axis resulting in an increase or decrease in pitching moment for a given angle of attack. A glider in equilibrium flight will gain a corresponding angular momentum, thus pitching up or down. Equation 18 shows that this change in pitching moment increases with tail volume and elevator surface area [5].

TABLE I
CONCEPTUAL DESIGN PARAMETERS

Parameter	**Purpose/Meaning**
Wings	Main source of lift
Airfoil	Wing cross section
Camber	Asymmetry of airfoil
Chord	Length of airfoil
Quarter chord	Point quarter of the chord away from the front of the airfoil
Wing span	Length of the wings
Aspect ratio	Numerical value that relates wing span in relation to the wing area
Stall angle	Angle of attack at which an increased angle decreases lift
Taper	Asymmetric chord length along the wing
Winglets	Wing tips pointed towards the sky
Sweep	Wings angled backwards towards the tail
Fuselage	Main body of plane
Twist/Incidence	Rotation of the wing in the direction of its elongation
Dihedral	Wings angled up/down in relation to fuselage
Tail	Lifting surfaces aft of the wings
Control surface	Surface that can be deflected in order to control plane
Elevator	Control surface on horizontal tail
Rudder	Control surface on vertical tail
Ruddervator	Combination of elevator and rudder

B. Controlling yaw

Controlling yaw is done with the rudder situated on the vertical tail, where the deflection δ_r is defined positive to the left. Analogously to the elevator, the addition to the yawing moment is a linear term in δ_r which increases with vertical tail volume V_V (defined in V-B) and rudder surface size [5].

C. Hinge moments

The control surface hinge moment coefficients are given by linear functions of deflection angle and angle of attack. For an elevator situated on the tail $C_{h_e} = C_{h_{\alpha_t}}\alpha_t + C_{h_{\delta_e}}\delta_e$, where α_t is the angle of attack of the tail. Calculating the derivatives in this linear relation often requires a wind tunnel [5]. However they can also be estimated numerically.

V. THEORY: CONCEPTUAL DESIGN

In this section the properties of wing and tail geometry are described, which has served as guidance for the choice of conceptual glider design. Table I displays a summary of the two following sections and the reader can find clarification from it when reading the remainder of the report.

A. Wings

The wings are the main source of lift for the glider, which makes determination of the wing dimensions an important step in obtaining a large enough glide ratio. The effect of different wing design characteristics are well known and the main ones considered in this project are described below.

The airfoil is the cross sectional geometry of the wings and it has important effects on lift and stall characteristics of the glider. Airfoil camber is a measure of the airfoils curvature. It akes the wing generate more lift than if the wing was completely flat or in general symmetric [2]. However, the more positive camber the airfoil has the more negative the pitching moment about the aerodynamic center becomes [2] [4], i.e. tendency for the glider to pitch down increases. Increasing airfoil thickness increases the drag of the wings [2]. At the same time, this also creates better stall characteristics. At stall, lift is generally lost quickly and the pitching moment changes drastically, but thick airfoils suppress these effects [2]. Evidently, there is a trade-off when selecting airfoils.

Large aspect ratio reduces the induced drag of the wings increasing the L/D ratio, but it also makes the wing heavier and lowers the stall angle [2]. In fact, $C_{D,induced} \propto 1/AR$ according to equation 2. This means it is more efficient to increase the wingspan b than the chord length c to achieve a given lifting wing area S. However, long thin wings are heavier than short and wide wings and even though they achieve higher maximum C_L they stall at lower angles of attack [2]. Trivially, long and thin wings do not have the same structural integrity as thicker shorter wings. This will ultimately put upper bound restrictions on the aspect ratio.

A tapered wing is when the root and tip chord are of non-equal length which can decrease the induced drag. Using a taper ratio of $\frac{c_{tip}}{c_{root}} = 0.45$ [2] increases the efficiency of the lift bearing wing as it resembles an elliptical wing which is proven to be most efficient. Another wing design technique to minimize the induced drag is the use of winglets. As discussed in section II-B2 induced drag is caused by escaping airflow over the wing tips. The use of a winglet, simply put, blocks the airflow and prevents it from passing over the tip [2].

Using wing sweep is a design choice often made for transonic and subsonic flight, as it limits the effect of the immense drag of the shock wave cone formed [2]. This drag evasion is attained by angling the wings backwards instead of them sticking out perpendicular to the fuselage (glider body, here the FFU).

Twisting a wing along its axis of elongation results in better control of stall location [2], which is desired by the pilot. Considering an autonomous solution, as in this project, incidence angle is of larger interest. It means using a uniform twist starting from the root which implies that the angle of attack of the wing is different from the one of the fuselage.

Angling the wings symmetrically upwards in the body axis system is referred to as using positive dihedral. This can improve the lateral stability of a plane as it restores roll instabilities. The amount of dihedral needed also depends on the vertical height the wings are mounted on the fuselage [2]. High mounted wings provide a natural dihedral effect. Too much dihedral however results in an unnecessarily oscillatory dutch roll mode [2].

B. Tail

An aft mounted tail is a way for gliders to achieve longitudinal and directional stability. A conventional tail consists of a horizontal tail wing (indexed H) that utilises the downwash from the main wings to create a restoring moment as a result of pitch up or down, and a vertical tail wing (indexed V) generating a restoring yawing moment due to sideslip [7].

Tails generally have lower aspect ratio than wings in order to stall at higher angles of attack than the wings, not losing pitch control at those angles [2].

A way to characterize the effectiveness of tails is the quantity called tail volume. The tail volume coefficient for a vertical V_V and horizontal V_H tail surface is given in the following expressions [2] [5]

$$V_V = \frac{L_V S_V}{bS} \tag{19}$$

$$V_H = \frac{L_H S_H}{cS} \tag{20}$$

where L_V, L_H generally are the distances between the wing quarter chord and the tail wing quarter chords, S_V and S_H the tail surface areas. Apart from their effects on control surface effectiveness, an important result is that the zero-lift pitching moment $C_m(\alpha_{L=0})$ and yawing moment derivative C_{n_β} grow with V_H and V_V respectively [5].

Tails come in different geometries and specifically the so called boom-mounted inverted V-tails can be used to avoid pusher propeller or jet engine wakes. The control surfaces on V-tails are called ruddervators. Deflecting both control surfaces up or down controls pitch while asymmetric deflections result in yaw as well as minor roll in the corresponding turning direction. In theory, V-tails generate less skin friction drag than conventional tails due to less total surface area being required to achieve the same projected horizontal and vertical areas. In reality however, roughly the same total area is needed to achieve the corresponding flight characteristics [2].

VI. Method

The methodology for the project can be summarised in the three categories: XFLR5 simulations [8], Simscale simulations [9] and prototype testing. The simulation software XFLR5 was used to test conceptual design ideas, and quick simulation runs allowed modifying designs through trial and error. In hope to verify the results from XFLR5 the authors continued in Simscale, a Computational Fluid Dynamics software (CFD), performing more sophisticated simulations of much longer runtime. To test the real flight capabilities of the designed glider while learning about RC airplane building the authors decided to finally build and test a real glider prototype.

A. Design requirements

The requirements serve as instructions for the glider designing process [2]. Although, not all requirements for the final BOOMERANG glider had been established, the following basic requirements were decided by the authors to be taken into account within the scope of this project.

1) The range of the glider should be contained in $[42, 66]$ km.
2) The altitude where the wings deploy from the FFU should be contained in $[5, 20]$ km.
3) The glider should be statically stable.
4) The glider should be dynamically stable.

These guidelines can be reformulated in pure aerodynamic terms as well as simplified to the most extreme case. The first two requirements (req. 1 - 2) are defined using large intervals. This motivated designing the glider with respect to an upper bound of the glide ratio when deploying from the maximum height possible. Deploying the wings at 10 km and covering a range of 66 km thus requires the glide ratio L/D = 6.6. This was taken as the result of req. 1-2. Req. 3 translates to requirements on the curves C_m, C_n and C_l as functions of α and β as described in section III-A. Req. 4 is fulfilled by req. 3 and having all aircraft modes in the negative half-plane of the root locus plots, as described further below in section VI-C.

Furthermore, a few special requirements were decided to not be followed exactly, only taken into consideration. This was done in hope to facilitate handling of them in the future while improving the design outcome of this project. These considerations were

- The wings and tail should, in stowed configuration, be stored inside a cylinder of diameter $= 240$ mm and height $= 30$ mm.
- The total mass of the glider and experiment should be 3 kg.
- The speed of the glider should not exceed a Mach number of 0.3.
- The glider's lifting surfaces should have adequate thickness in order to attain structural integrity.

This meant that the very complex glider design problem could be reduced to the goal of finding the smallest wing and tail configuration that completes the requirements above and lifts a 3 kg glider at speeds corresponding to incompressible flow (see section II-E). The speed consideration simplifies calculations since the dependence on Mach number can be disregarded. Furthermore, a certain thickness is needed for the wings to not break, but structural analysis is rather the focus of project I2. Wings with minimum thickness larger than 10 mm was considered reasonable.

B. Initial calculations

Having small wings however meant that the glider needed to fly faster in order to achieve the same lifting force, which can be seen in equation 9. The speed of the FFU at 10 km was of interest, since assuming equilibrium free fall this was an upper bound of velocity of which the glider could be accelerated to solely by the force of gravity. Regardless, this would roughly be the initial velocity of the glider the moment the wings deploy.

Although the REXUS rocket and the FFU have different drag coefficients, the terminal velocity of the FFU at 10 km was approximated as the velocity of the rocket at that altitude. The REXUS user manual [10] includes rocket kinematic data

from the SPIDER-2 experiment. This shows that the falling speed of the rocket at 10 km is 100 m/s, which conveniently amounts for a Mach number of $100/299.5 = 0.33$ [2]. The gliding speed will be slower than the free-fall speed. Due to the rough approximation above it is reasonable to assume that the glider will stay under 0.3 Mach during gliding flight, thus consistent with the consideration in the previous section.

The next step was to look at historical aircraft design concepts in [2], using historical data and suggestions presented in the book as a starting point for glider design. The characteristics of the wing and tail considered are defined in table I.

C. Using XFLR5 for glider design

XFLR5 is a simulation tool with short runtime that uses potential theory to estimate lift, drag, moments and velocities of RC airplanes, with stability analysis integrated into the software. This meant that initial design could be made seamlessly without the hassle of using full CFD programs early in the design process. Modelling the fuselage in XFLR5 was however an issue due to limitations of the program. To get the correct inertia from the main body point masses were distributed in the volume occupied by the FFU. The estimated drag of the FFU was also added. The method of design testing meant that a glider had to pass certain stages of XFLR5 based on the design requirements in section VI-A above. If more than one glider design passed the stages they could be compared.

1) Stage 1 - Longitudinal static stability: Pitch moment coefficient fulfills constraints in section III-A1.

2) Stage 2 - Glide ratio: The glide ratio is larger than 6.6, based on the requirements in section VI-A.

3) Stage 3 - Velocity: The flight speed of the glider does not pass Mach 0.3 as mentioned in section II-E.

4) Stage 4 - Dynamic stability: All poles attained after running a stability analysis have a real part less than zero. Exception can be made for the spiral mode, if it is slightly positive.

5) Stage 5 - Remaining static stability: Roll and yaw moment coefficients fulfill constraints in section III-A2 and III-A3 after running a sideslip batch analysis.

D. Failure of design in XFLR5

Assuming a glider solution failed a given stage above certain measures needed to be taken in order to improve the design. Each failure would result in the following actions:

1) Failing stage 1: If $\frac{\partial C_m}{\partial \alpha} < 0$ was not fulfilled the center of gravity was moved to alter the slope.

If $C_m(\alpha_{L=0}) > 0$ was not fulfilled either the horizontal tail volume V_H was made larger to combat the negative pitching moment or the effect of the cambered wing was reduced.

2) Failing stage 2: If the glide ratio was not sufficient more lift was needed. This was created with different methods such as increased wing area, more cambered airfoil or finer methods like using taper ratio. The center of gravity could also be moved to assure that the maximum glide ratio occurs for steady flight (when $C_m = 0$).

3) Failing stage 3: If the glider was too fast the main solution was to increase the wing area. By having more lifting surface a slower velocity would produce the same net lift.

4) Failing stage 4: If the lateral modes were unstable, dihedral was used to force them into the left half-plane in the root locus diagram. Specifically if the directional mode was unstable a larger vertical tail could solve the issue.

5) Failing stage 5: If $\frac{\partial C_l}{\partial \beta} < 0$ was not fulfilled more dihedral was added to the wings.

If $\frac{\partial C_n}{\partial \beta} > 0$ was not fulfilled a larger vertical tail volume V_V would usually solve the directional issue.

E. Data for other I context groups

As the project groups of Context I are closely intertwined data was often shared with groups I2, I3 and I4. I3 used the lift, drag and moment coefficients of the the different models to simultaneously test their controller on the glider solution as it was in its design stage. The dynamic stability calculated by I3 was also compared to the stability retrieved from stage 4 of the XFLR5 design process. I4, on the other hand, are in charge of the electrical systems of the glider and were given information about the time of flight and hinge moments as discussed in sections II-F and IV-C. Last but not least, glider dimensions and velocity was shared with group I2.

F. CFD

As XFLR5 uses potential theory its results are an approximation, but that is also the case when using CFD software. The difference is that a CFD program numerically solves the fundamental Navier-Stokes equations to retrieve the flow data of a system. On the other hand XFLR5's use of potential theory produces less reliable data since it does not utilise those equations. Another limitation of XFLR5 is its inability to account for the fuselage. This is an acceptable shortcoming for planes with a low profile fuselage, but not for the large FFU in this project. The dilemma for the authors was to conduct a fast initial iterative process of the model by using XFLR5 but still using trustful data. The work around was for the authors to use the CFD program Simscale at a final stage of the model with the goal to either validate or disqualify the results acquired from XFLR5.

1) Acquire drag of FFU: Primarily the drag of the cylindrical FFU was calculated in Simscale. The reader is reminded of the issues of simulating the fuselage in XFLR5. This FFU drag was then input as extra drag for the glider model in XFLR5 which resulted in more realistic data, especially glide ratio values.

2) Full CFD: When a final glider solution had been established in XFLR5, with design features corresponding to the list in section VI-B, a CAD (Siemens NX) model was created. This model was then used in a virtual wind tunnel created in Simscale to retrieve fluid dynamic data to be compared with XFLR5 and real life tests.

G. Simscale

One can relate the use of Simscale as a virtual wind tunnel and many parameters were considered during set up. The CFD was set up according to the professional tutorials [11] and [12].

1) Flow type: Incompressible flow was used as XFLR5 confirmed that the Mach number was below 0.3 and the conditions of section II-E were fulfilled.

2) Flow speed: The flow speed was also chosen according to XFLR5, where the speed of steady state flight was of interest. That means that the speed when $C_m = 0$ was used in Simscale. In the case of the final glider design of this project, the speed was 28 m/s at sea level.

3) Angle of attack: When setting angle of attack one has to remember that Simscale works as a wind tunnel, which means that the velocity vector has to be considered with the magnitude equal to the speed defined above. That meant that the speed parallel and perpendicular to the glider had to be altered in Simscale according to

$$U_{\parallel} = Ucos(\alpha) \tag{21}$$

$$U_{\perp} = Usin(\alpha) \tag{22}$$

where U and α is the desired speed and angle of attack

4) Region refinements: Areas of specific interest and possible complexity demand extra attention. This was done through region refinements that are small areas of detailed meshing which means that the Navier-Stokes equations are solved in smaller geometric steps there. In the case of the final glider region refinements were applied to the surface of the glider and the wake behind it.

H. Prototype testing

The following section describes the methods used for building the prototype.

1) Scaling: In order for a prototype glider to demonstrate the same flight characteristics as the final glider descending in the atmosphere, the two cases needed to be invariant with respect to the similarity parameters [3]. Due to incompressible flow requirement $0 \leq M < 0.3$ the Mach number dependence can be completely disregarded [3], and focus only needs to remain on the Reynolds number. Test flying the prototype at sea level then reduces to only altering the size and/or free flow speed to acquire the same Reynolds number as at a specific height of interest.

In this project the built prototype was in scale 1:1 as it corresponded to dimensions and velocities of regular RC planes. If it would have been a smaller scale, the flight speed would have needed to be increased to maintain the right Reynolds number. Also, space for electronics to move the control surfaces would have been limited.

2) Blueprints: A more detailed CAD of the glider exterior was first made, serving as a blueprint for manufacturing. This design utilised the conventional spars and ribs method of making wings and tail interior, reducing weight while still maintaining internal structure. A shrink film was then used as the surface of the wings, which was attached and shrunk down using a heat gun. Furthermore hollow booms connected the tail with the FFU fuselage, and was mounted atop the FFU. The FFU itself was modelled as a simple cylinder of corresponding dimensions.

The overall design of the glider was kept the same as in XFLR5 but the tail was changed slightly to simplify building. The ruddervators were realised using 100 % of the chord rather than the outcome of XFLR5 analysis. The latter allows for supreme control effectiveness [2] and equation 18. It also makes for a simpler tail design, with the whole chord rotating about the same axle it is mounted. On top of the tail a boom connector was made to prevent collision when pitching down. Also the booms were mounted atop the fuselage instead of through holes in the FFU wall.

Additionally, since small deployable wings were desirable, the glider needed to fly fast and changes were thus made to allow a propelled start from ground. A motor and motor mount were added aft of the FFU, pointed downwards in order for the pusher propeller to fit in between the tail booms as well as to not disturb the flow around the wings and tail. (Note that the addition of the propeller technically made the glider into a model RC airplane, but since gliding capabilities is the main interest, the authors will still refer to the aircraft as the glider). Propeller ground clearance forced the addition of landing legs mounted on the bottom of the fuselage. Apart from the ruddervators, small ailerons were also added to the tips of the wings to hopefully allow some roll control during flight in case needed. These additions were all suggested by BOOMERANG team member Victor Nan Fernández-Ayala (VNF). However, none of them were intended for the final design.

3) Materials and components: The components of the glider were manufactured in carbon fibre, Plexiglas, 3D printed plastic and shrink film. The parts were either screwed on or super glued using accelerator spray. The carbon fibre was only used in rods of various inner (I) and outer (D) diameters, in the wing spars (I = 4 mm, D = 6 mm), tail spars (solid, D = 2 mm) and tail booms (I = 8 mm, D = 10 mm). The Plexiglas was used for the hollow FFU cylinder (bent using a large heating gun), wing and tail ribs, motor mount and landing legs. The top lid of the cylinder had two hinges that allowed opening up the fuselage and reaching the electronics inside. The Plexiglas had a thickness of 5 mm and all parts were cut out using laser cutter. Last but not least, 3D printing was used to construct parts of more complex geometries. These were an upper connector for the two tail surfaces, two lower connectors for the tail surfaces and the booms, rods in an inverted T configuration stabilising the tail, two connectors for the tail booms to the fuselage, two L shaped arms connecting the servos with the ruddervators, two connectors for the wing roots to the exterior of the fuselage, stabilising connectors for the wings on the interior of the fuselage, mounts for the aileron servos on the interior and four wheels attached to the landing legs. Also, four short hollow cylinder stops were glued on the ends of the carbon fibre tail axles holding the V-tail surface together while still allowing the ruddervators to rotate.

4) Electronics: The RC airplane electronics including flight computer, GPS, radio control receiver and the two aileron servos were hot glued inside the fuselage, and the ruddervator servos directly on the tail. The motor for the propeller was attached on the aft motor mount and the heavy battery in the front part of the fuselage which placed the center of gravity in roughly the position corresponding to the simulations. Cables connecting the tail servos were led through the tail booms and

the hollow hinge gap in the lid. All cables were soldered to the flight computer. Soldering was mainly done by VNF, who also had prepared the purchase list for the electronics. VNF also attended during most of the manufacturing period, providing aid with various parts of the glider build.

I. Ground & flight test

After finishing the glider build, a ground and flight test was conducted. During the ground test, the electronics were calibrated to give equal and adequately large deflections in both directions, and the motor was tested in order to move the glider on asphalt. After completing the ground test the authors together with VNF, another BOOMERANG team member and a spectator drove to an open field near Stockholm, Fisksätra to conduct the first test flight. The method of testing was to start the glider from a gravel road and accelerate using the propeller. Then to pitch up and try to lift off the ground, after which the glider would quickly be brought down again. If successful, maneuverability in all directions and gliding after turning off the propeller would be tested during a more prolonged flight. The idea was then to approximate the glide ratio by eye and camera footage. The flight test ended with a drop test from a cliff, simply hand throwing the glider as modelled in the simulations i.e. without motor and landing legs.

VII. Results

”Berg”, in Swedish, means mountain. A mountain is a large piece of rock that would fall quite fast through the atmosphere and that is the inspiration behind the name of the final glider design. Since it is final the authors decided to ”freeze” it. Hence the final design name became ”ICEBERG”.

A. ICEBERG: The ”frozen” glider design

What led ICEBERG to be the frozen design is its combination of desirable geometry and aerodynamic properties. The glider has rectangular wings without ailerons and a boom-mounted inverted V-tail with two ruddervators. ICEBERG modelled in CAD can be seen in figure 6 and figure 7. The dimensions of the lifting surfaces are listed below.

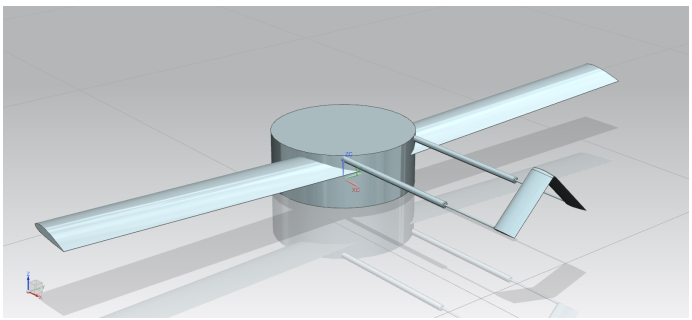


Fig. 6. 3D backside view of ICEBERG made in CAD. The boom mounted inverted V-tail is clearly visible, as well as the wings. This 3D model was used for CFD simulations in Simscale.

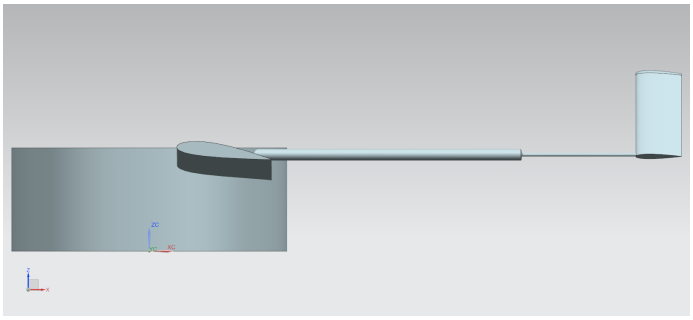


Fig. 7. 2D left side view of ICEBERG, showing the dihedral and incidence of the wings.

1) Wings:

- airfoil NACA2415
- span $b = 900$ mm
- chord $c = 83$ mm
- area $S = bc = 0.0747$ m^2
- aspect ratio $AR = 10.843$
- taper ratio $\lambda = 1$
- dihedral $\Gamma = 2°$
- high mounted wings
- incidence $i = 7.5°$

2) Inverted V-tail:

- airfoil NACA0015
- span $b_t = 200$ mm
- chord $c_t = 40$ mm
- area $S_t = b_t c_t = 0.008$ m^2
- aspect ratio $AR_t = 5$
- taper ratio $\lambda_t = 1$
- moment arm $L_t = 390$ mm
- $45°$ angle between each tail surface and the vertical plane
- incidence $i_t = 0°$

3) Ruddervator:

- length $l_{r-e} = 80$ mm
- chord $c_{r-e} = 60\%$ of $c_t = 24$ mm
- maximum deflection $\delta_{r-e_{max}} = 20°$ in both directions

B. ICEBERG in XFLR5

The XFLR5 plane view can be seen in figure 8.

Considering the aerodynamics, ICEBERG fulfills all stages of section VI-C except for stage 2 which is proven by figures 9, 10, 11, 12, 13 and 14. Specifically one can see in figure 12 that the spiral polar is slightly positive, not contradicting stage 4 in the XFLR5 methodology in section VI-C.

Figure 9 shows that the pitching moment decreases with increased angle of attack, figure 10 shows a decreasing glide ratio versus pitching moment with a maximum glide ratio around 4.7, figure 13 shows a decreasing rolling moment versus sideslip angle and figure 14 reveals an increasing yawing moment with an increasing sideslip angle.

Figures 11 and 12 show that all polars are in the left half plane except for the spiral polar as described above.

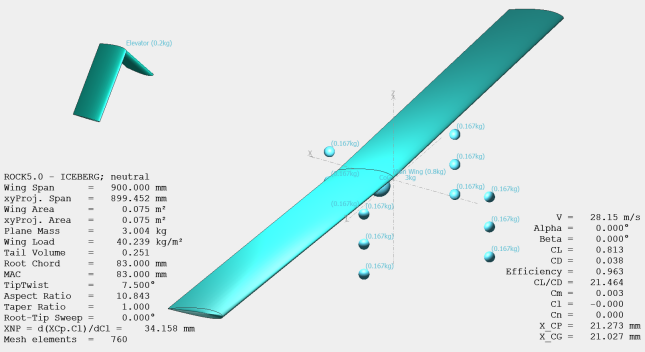


Fig. 8. 3D view of ICEBERG in XFLR5 at $\alpha = 0°$. Specifications calculated by the software is printed in the bottom corners. Also, point masses at the boundaries of the FFU are visible. The desired location of the center of gravity, can barely be seen as a large sphere under the wing.

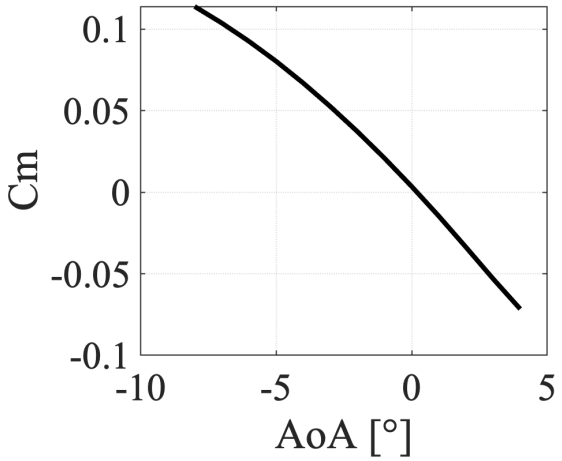


Fig. 9. Pitching moment coefficient vs angle of attack, ICEBERG in XFLR5. Plot conversion into IEEE standard from [13]

C. ICEBERG in Simscale

When studying ICEBERG in Simscale figure 15 shows that the pitching moment increases with angle of attack compared to a negative slope in XFLR5. Figure 16 shows a parabolic function of glide ratio versus pitching moment with a much lower maximum value than what was obtained in XFLR5. Finally figure 17 shows a slightly increasing drag with increasing angle of attack that is larger than what is found in XFLR5.

D. Separate FFU in Simscale

Data retrieved in Simscale from looking at the FFU separately reveals in figure 18 that the drag of the FFU is a parabolic function versus angle of attack with a minimum at zero angle of attack. Figure 19 shows that the pitching moment increases with increasing angle of attack for the FFU.

E. XFLR5 setup in Simscale

Using the same model in Simscale as in XFLR5, which is only the wings and tail, figure 20 reveals a decreasing pitching moment with increasing angle of attack. The figure also shows that this is similar to the values retrieved in XFLR5. Figure 21 points to an exponential behaviour of the glide ratio versus pitching moment.

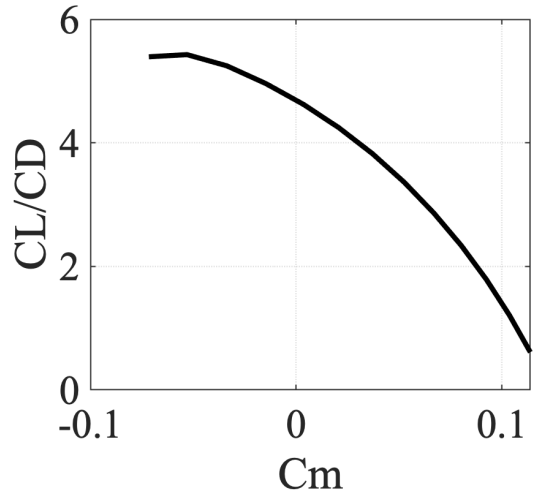


Fig. 10. Glide ratio vs pitching moment coefficient, ICEBERG in XFLR5.

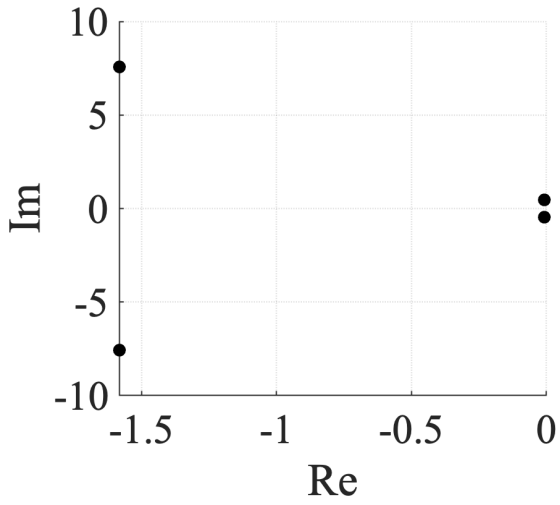


Fig. 11. Longitudinal dynamic polars (Imaginary axis in $rad\ s^{-1}$ & Real axis in s^{-1}), ICEBERG in XFLR5.

F. ICEBERG 383X

Before manufacturing ICEBERG it was realised that the tail was completely in the wake of the FFU. A sub model was created, called ICEBERG 383X which represents a tail volume 3.83 times larger than the original ICEBERG. Figure 22 shows the turbulent kinetic energy [14] around the tail.

G. The glider prototype

The CAD serving as the main blueprint for manufacturing the glider can be seen in figure 23. This shows the spars and ribs configuration for the wings and tail, as well as some of the 3D printed components of more complex geometry.

Figure 24 shows a picture of the manufactured glider. It utilises the same glider dimensions as ICEBERG 383X, however also with a tail mounted slightly higher than the original ICEBERG since it was mounted on the top of the FFU lid. In figure 25 a closer view of the tail with the servos and servo-arms connected to the ruddervators can be observed. The total mass of the glider with all electronics was 2.42 kg.

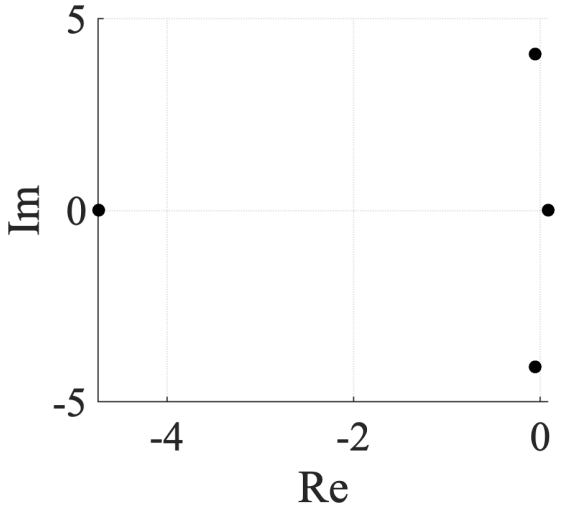


Fig. 12. Lateral dynamic polars (Imaginary axis in $rad\ s^{-1}$ & Real axis in s^{-1}), ICEBERG in XFLR5.

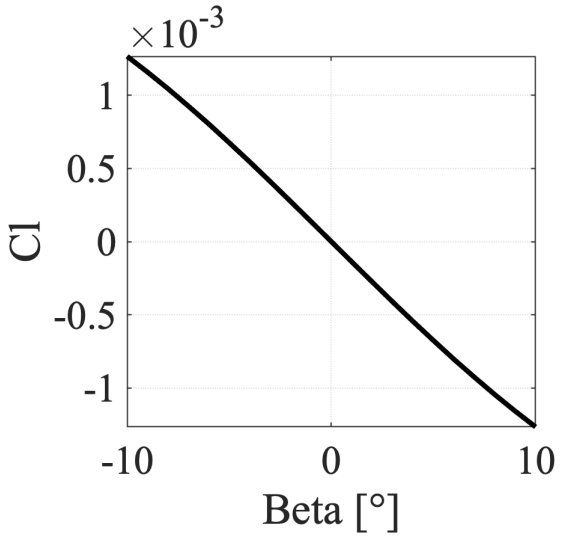


Fig. 13. Rolling moment coefficient vs sideslip angle, ICEBERG in XFLR5.

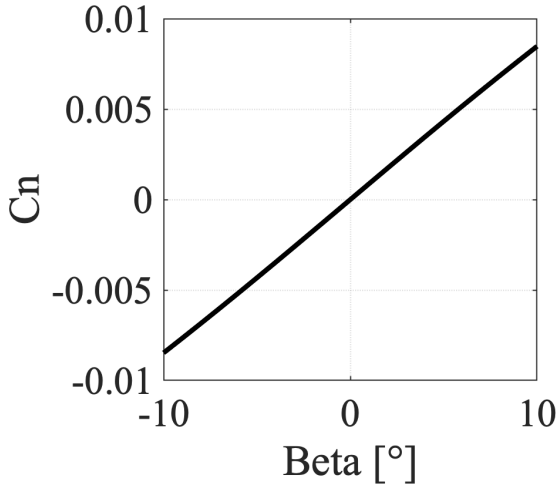


Fig. 14. Yawing moment coefficient vs sideslip angle, ICEBERG in XFLR5.

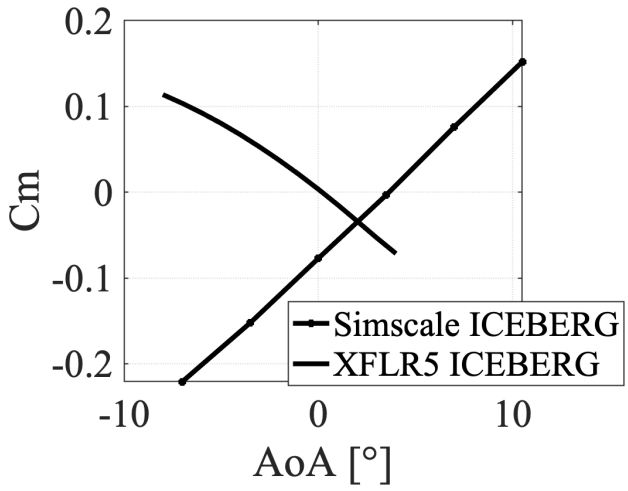


Fig. 15. Pitching moment coefficient vs angle of attack, ICEBERG in Simscale and XFLR5.

H. Ground test

After calibration, both the ailerons and the ruddervators were able to complete equal and medium deflections both up and down. The propeller was also able to rotate.

The tail surfaces wobbled back and forth when deflecting the ruddervators, but the wobble was significantly reduced when attaching the T shaped stabiliser seen in figure 25. The ailerons could easily be disturbed and deflected by hand.

Driving on asphalt, the glider was able to slightly yaw left and right using thrust from the propeller and deflecting the ruddervators correspondingly. The turn radius was however not optimal, and the glider sometimes turned to the right even without extending any control surfaces.

I. Test flight

The flight test unfortunately did not give much information about the flight capabilities of the glider. Starting from the gravel road and using about 90 % power of the propeller, the glider started to accelerate after which the wheels lost grip and it began to slide with its right side facing the direction of travel. This motion eventually made it tip over. The glider crashed on the side of the road never having lifted entirely.

Many parts were damaged in the crash. Due to the load sideways, all three landing legs broke off. The motor mount also came loose on impact, and the propeller tips broke so that it was only half as long. Only one tail servo came loose. The rest including the electronics held together with hot glue. The carbon fibre rods were so strong that the long extended wings and tail looked untouched. The film was punctured in only one place.

Since the FFU fuselage, wings and tail were not severely damaged, the drop test could still be performed. This however did not result in much glide. The glider simply fell to the ground travelling in what seemed to be a regular ballistic trajectory. After two drops, only the wings and tail booms were intact.

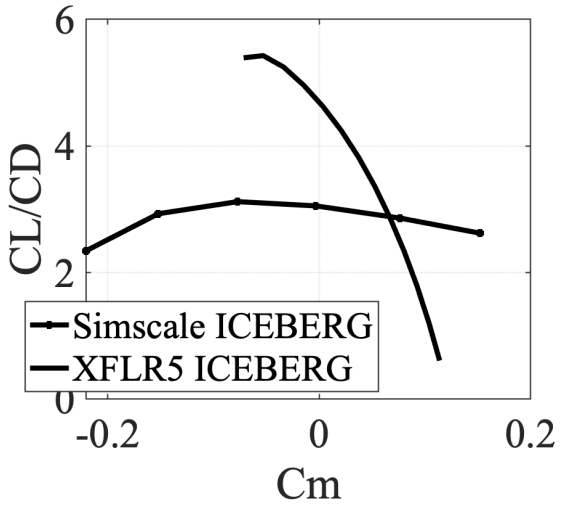


Fig. 16. Glide ratio vs pitching moment coefficient, ICEBERG in Simscale and XFLR5.

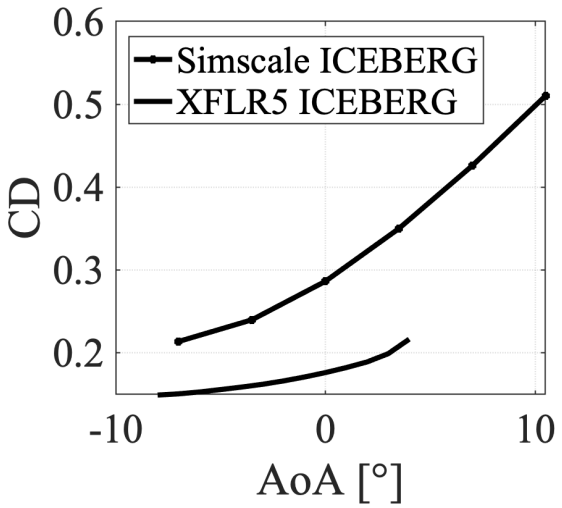


Fig. 17. Drag coefficient vs angle of attack, ICEBERG in Simscale and XFLR5.

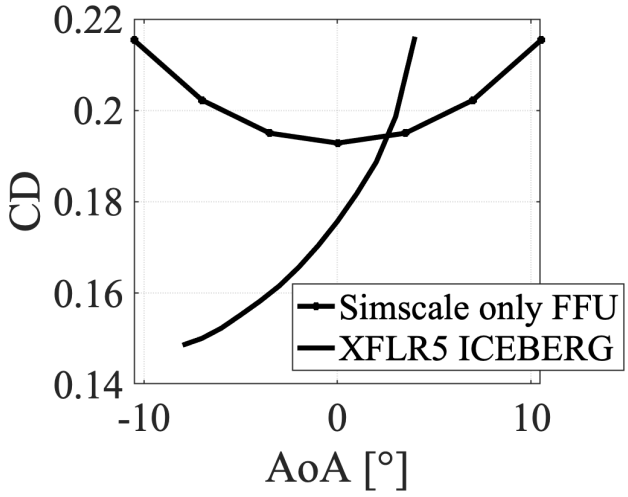


Fig. 18. Drag coefficient vs angle of attack, only the FFU in Simscale and ICEBERG in XFLR5.

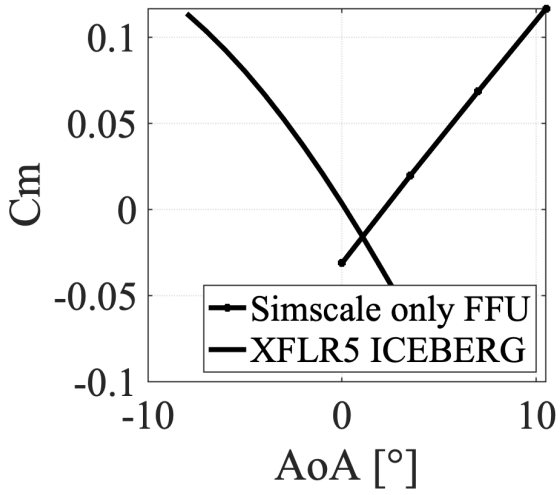


Fig. 19. Pitching moment coefficient vs angle of attack, only the FFU in Simscale and ICEBERG in XFLR5.

VIII. Analysis & Discussion

A. Why ICEBERG?

The selection of dimensions for ICEBERG after XFLR5 was motivated by known effects described in the theory sections. However, no optimization has yet been performed. The unseen effects from CFD analysis was assumed to affect the flight results more in the end, meaning that optimization at this stage would be a waste of time. This is also consistent with conceptual design described in [2].

Referencing back to the list in VII-A and looking at figures 9-14, the following motivations were made:

1) Wing decisions: NACA2415 was chosen as the airfoil of the wings. The camber creates some additional lift and not enormous pitch down moment. From lift coefficient diagrams in XFLR5, the authors concluded that using a non-zero camber still increases the lift of the aircraft enough to motivate using it as opposed to a symmetric airfoil. A more cambered airfoil such as NACA4515 would require a larger horizontal tail volume and consequently a larger tail more difficult to deploy. This is simply not worth the extra bit of lift gained. The thickness of 15 % was due to historical data in [2]. The ratio is also sufficient for the thickness consideration of ≥ 10 mm. The airfoil choice can be perfected at a later stage. Raymer [2] suggests not getting stuck at airfoil choice.

Deciding the size and dimensions of the wings required extensive experimenting and many trials. The main complication was that the wings needed to have a certain lifting area in order to generate enough lift to compensate for the enormous relative drag of the FFU (figure 18) which brought down the glide ratio by a lot.

Figure 10 shows the glide ratio C_L/C_D versus C_m from XFLR5. This is close to a monotonically decreasing function, different to what could be seen without using the extra drag from the FFU. C_m is in turn decreasing function of α, seen in figure 9. At too large α the wings stall, so that region simultaneously had to be avoided. Consequently, the best range of the glider would require flying at the largest possible angle of attack. A reasonable value was thought to be $7.5°$.

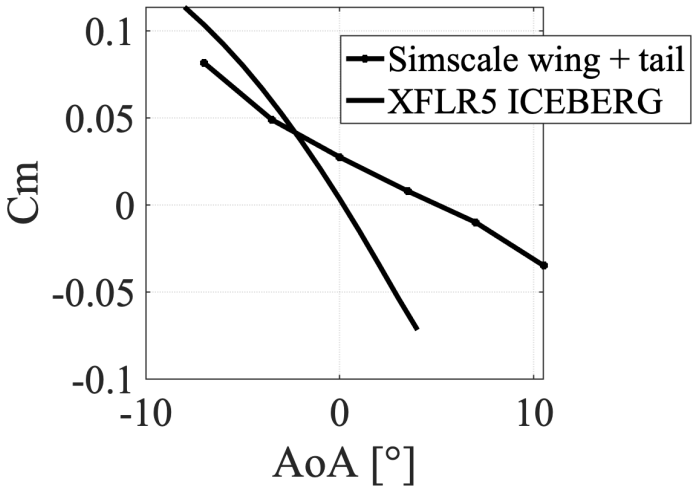


Fig. 20. Pitching moment coefficient vs angle of attack, wing + tail in Simscale and ICEBERG in XFLR5.

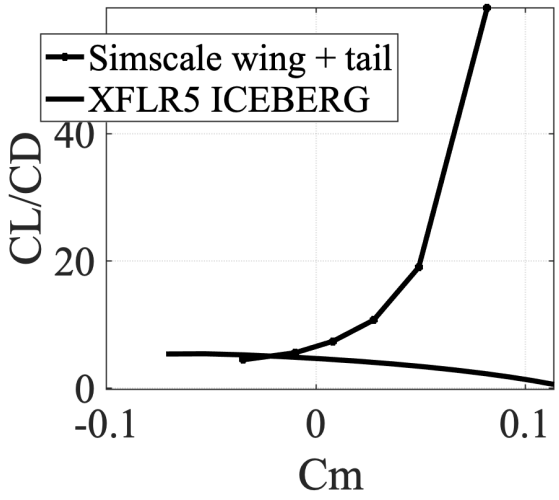


Fig. 21. Glide ratio vs pitching moment coefficient, wing + tail in Simscale and ICEBERG in XFLR5.

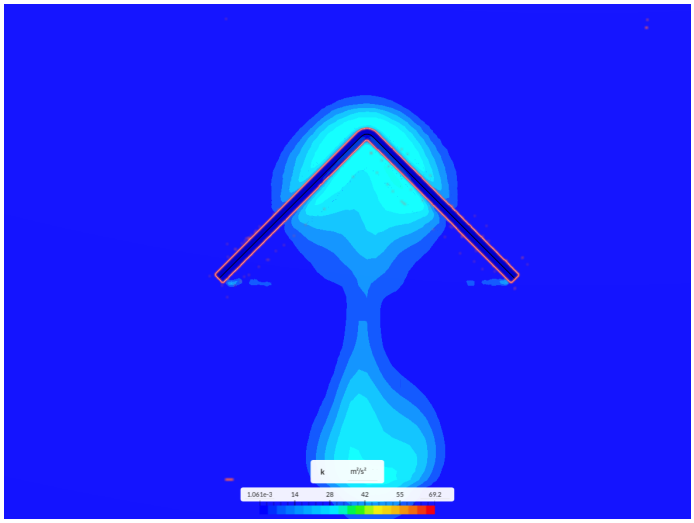


Fig. 22. Simscale data of turbulent kinetic energy around the tail of ICEBERG 383X.

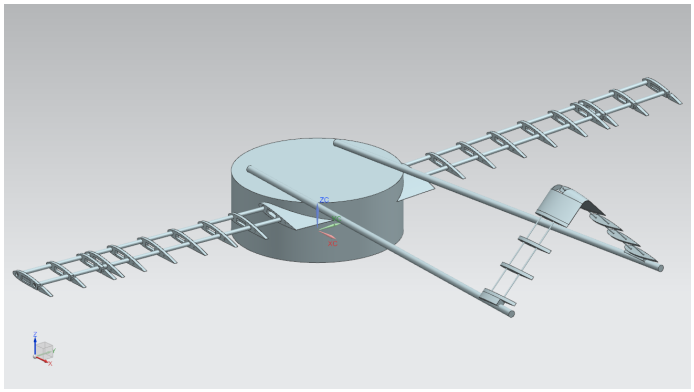


Fig. 23. Backside 3D view of ICEBERG 383X designed in Siemens NX. Spars, ribs, tail and wing connectors can be seen in the image.

The wing dimensions listed in section VII-A was primarily thought to result in a glide ratio of 7.6 at steady flight. However, according to figure 10 one sees that the glide ratio is in fact 4.7, which means that it fails stage 2 of section VI-C. This was a misfortune due to an incorrect FFU drag used in XFLR5 and was discovered during manufacturing. The incorrect drag came from a misuse of Simscale, where a symmetry plane setting yielded results half of the correct values. Instead of changing the glider design last minute, the authors accepted the lower glide ratio for now, as it was still contained in the requirement interval. Future designs will however aim to satisfy the upper bound glide ratio requirement of 6.6.

The small chord length yields a large horizontal tail volume due to equation 20 while simultaneously allowing a wing thickness of 12.5 mm, slightly greater than what was assumed to be the bare minimum (10 mm). This meant the projected horizontal tail surface and moment arm could be made as small as possible to achieve a requested tail volume. The span was then sized to obtain sufficient glide ratio. This effectively maximized the aspect ratio, and mimics the high performance wings of sail planes.

The main complication is that the large span decreases the vertical tail volume, which demands for increasing the projected vertical tail surface instead. However, the yawing moment contribution from the FFU is assumed to be negligible compared to its pitching moment contribution which in figure 19 is shown to be very destructive. Keeping the horizontal tail volume large is thus the limiting factor for ICEBERG.

Unfortunately, the longer the wings, the harder they are to deploy. Furthermore, tapered wings would be harder for group I2 to construct a deployment mechanism for, which resulted in the choice of rectangular wings. If the current wing span would be a problem for deployment, the wings can be shortened. However, then the chord must increase to compensate the loss in lift, and tail made larger to counteract the loss in horizontal tail volume. Sail planes normally have even larger aspect ratios than ICEBERG, but one must remember those wings need not be deployed mid-air.

An incidence angle of 7.5° made it possible for the glider to achieve the requested glide ratio when the FFU experienced minimum drag, a way of design suggested in [2]. As predicted,

Fig. 24. The manufactured glider during the ground test outside the student workshop at KTH Royal Institute of Technology. Wings, tail, fuselage and landing gear is visible. The battery sits on top, but was placed inside the fuselage before the flight test.

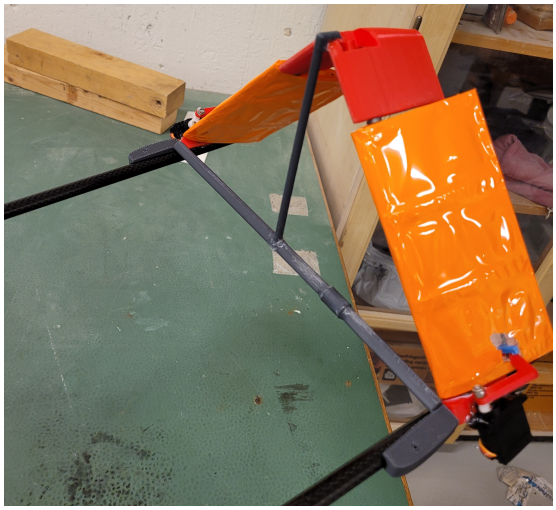


Fig. 25. The manufactured boom mounted V tail. Both ruddervators are deflected corresponding to a left yaw maneuver.

figure 18 shows that the FFU contributes to minimum drag at $\alpha = 0$. At that angle of attack, the wings still experience $\alpha = 7.5$ due to the incidence.

The dihedral of 2° did two things: First, it provided the glider with static roll stability as seen in figure 13. Second, it made the two modes closest to the imaginary axis move over to the left half-plane of the longitudinal root locus plot after performing stability analysis, visible in figure 11. This might differ from the real dihedral achieving this stability, since high mounted wings like these have an additional dihedral effect. To not excite the dutch roll mode, dihedral lower than 2° might be sufficient. Test flights are needed to analyse this further.

Using 11 with the sink rate at 5 km altitude, the time of flight from 10 km to sea level is estimated to 24 minutes. With the desired glide ratio of 7.6 the time is instead 38 minutes. These values seem reasonable for a returning glider, and is comparable to a descending airplane by order of magnitude.

2) Tail and control surface decisions: The tail airfoil was chosen as NACA0015, inheriting the thickness of the wings for consistency. Furthermore it is uncambered, which usually is the case of tail airfoils. It is common to put negative incidence on the tail to increase the restoring pitching moment [2], but after having tested conventional tails in XFLR5 and seeing that it hardly made any difference, the authors accepted a 0° tail incidence.

The inverted V-tail was used in hope to avoid the turbulent FFU wake, hindering the tail from getting blanked inside it and thus maintaining control [2]. A conventional tail would possibly require a much higher placement to not get blanked in the wake, increasing drag due to the leverage arm extended upwards, also creating a small but unwanted pitching moment. Assuming a cylindrical wake extending aft of the FFU, an angle of 45° between the tail surface and the vertical plane was seemingly a good option for wake evasion.

The tail span, chord and moment arm were then chosen to achieve static longitudinal and directional stability which can be seen in figures 9 and 14. Trial and error led to the dimensions listed in VII-A. For simplicity no taper was used and the aspect ratio of 5 is far less than the wings to prevent tail stall.

Last but not least, the dimensions of the control surfaces were chosen to see an increase in yawing and pitching moment when deflected. equation 18 implies a very effective tail control surface is using all of it as a ruddervator. ICEBERG uses 60 % of the chord which is enough to change the pitch and yawing moments acting on the glider, thereby controlling it. Increasing this fraction to up to 100 % is possible but the built prototype glider showed that this is not optimal (see section VIII-E). Furthermore, the 60 % control surface size yields low hinge moments. $\alpha = 10^\circ$ and $\delta_{r-e} = 20^\circ$ represents an edge case of what the glider most likely will experience during steady flight. Analysis in Xfoil of this orientation showed that the hinge moment on a single control surface is in the order of 0.1 Nm, which there exists motor solutions for according to group I4.

B. ICEBERG and Simscale

Unfortunately the data retrieved from Simscale did not provide a correct picture of the aerodynamic properties of ICEBERG. Looking at figure 15 and 16 one can observe large differences in pitching moment and glide ratio. Figure 15 shows that it is even unstable in Simscale! Primarily the idea was that the issue emerged from the absence of the cylindrical fuselage in XFLR5. But when considering the same model as in XFLR5, without the unconventional cylindrical fuselage, figure 20 and 21 reveal that the issue rather lies in the setup of the actual CFD program. These figures should actually prove, more or less, an identical pitching moment effect between XFLR5 and Simscale. At first the Matlab code that plotted the data was scrutinized for incorrect implementation, but no error was found there.

Secondly, the drag of ICEBERG in XFLR5 and Simscale should coincide. Remember that ICEBERG in XFLR5 has

an added drag term for the cylindrical fuselage (FFU). But figure 17 reveals that is not the truth and further points to an implementation error.

Even though no qualitative data can be trusted from Simscale one main hypothesis is quantitatively proven: The FFU adds significant drag and a large pitching moment that increases with angle of attack. This can be seen in figures 18 and 19.

A future solution to the inconsistencies in Simscale is a larger investment into CFD resources. That could be realised through more time in Simscale, trying another software and/or receiving experienced advice.

C. ICEBERG 383X

Before continuing with manufacturing, the authors created the sub model of ICEBERG due to the discovery above and the fact that the FFU creates a large aft wake. Figure 22 shows how the larger tail of the 383X sub model is mostly outside the wake of the FFU, as areas of low turbulent kinetic energy are little affected by the wake. This creates better maneuverability and since the tail volume is larger also combats the large pitching moment of the FFU.

D. Manufacturing improvements

During manufacturing of ICEBERG 383X one of the main difficulties was to attach the film to the ribs of the wing structure. The heat gun did not provide even heat distribution and the film buckled in between the ribs, especially around the small ailerons. Overall it did not form a smooth cover which made the resulting airfoil of the real model inconsistent with the chosen NACA2415 airfoil. For the future, the more conventional heating iron should be used instead of the heat gun combined with a better spar and rib structure, adding additional spars around the leading and trailing edge in order for the film to successfully attach around those corners. Another solution would be to make the wings entirely in carbon fibre, using molds to shape the exterior airfoil.

E. Ground test improvements

Although using 100 % of the chord as ruddervators makes for effective control, it also creates large instability in the tail. The configuration requires the whole tail surface used for the ruddervator to be mounted on a single rotating axle. Additionally, the ground test showed that these ruddervators could easily be disturbed and deflected unwillingly. The servos apply forces counteracting undesirable rotation due to disturbances, but the longer the chord the better the leverage arm, enabling disturbances to deflect the ruddervators regardless. Vibrations and wind would most likely be sources for this. Instead, using < 100 % cord allows the leading tail surface to be attached fixed and a trailing control surface of less leverage arm to rotate about a single axle, possibly with less deflection due to wind and vibrations.

This was how the ailerons were constructed, however the 2 mm axle on which they were mounted on was very flexible which made them easy to move by hand. Just a small torsional load caused the ailerons to deflect even though the servos did not move. This could be solved with an axle of larger diameter.

No attempt in improving steering on ground was made as this would not affect the final glider design.

F. Flight test improvements

Since the prototype glider crashed before lift off, improvements need to be made in order to make it flyable next time. The glider needs to acquire a speed according to equation 9 to glide at a specified glide ratio. This ”barrier” can be avoided for the final glider design since falling through the atmosphere will already provide it with sufficient kinetic energy, but for repetitive testing this is a problem. Again, equation 9 shows that a possible solution is making the surface area of the wings larger, which would decrease the speed at which sufficient lift is attained, allowing the glider to fly at slower speeds. This is already desirable since the obtained glide ratio for ICEBERG was not sufficient anyway.

However these wings might still not be enough to lift off the ground at reasonably low speeds and as discussed before making the wings too large is contradictory to the requirements, which ask for small and deployable wings. Luckily, equation 9 reveals that the glide speed can also be reduced by decreasing the mass of the glider. For testing purposes only, it might thus be of interest to test fly a lighter glider than 3 kg. This was the reason why the mass of the manufactured glider was left to be whatever the total mass of the parts were, in this case 2.42 kg. What effects reducing the inertia has on other flight characteristics need to be examined before implementing this method again in the future.

In order for the propeller to lift the glider off the ground, the landing legs need to be more durable and mounted to prevent sliding and instability. The back legs could be attached at an angle to provide larger distance between the rear wheels. Four legs could possibly be used instead of three to provide even more stability.

The height of the legs should also be reduced, which would require the propeller to be mounted higher up. This is possible without a double boom mounted inverted V tail which restricts usable area aft the FFU. Instead, both control surfaces could be attached to a single boom connected to the middle of the inverted V. This would cause a small increase in drag and alter the pitching moment since that boom would need to be mounted with an angle upwards. Most importantly, this design might even be simpler to deploy.

Both the landing legs, the FFU, and the motor mount were made in plexiglas, which the flight and drop test showed is very brittle. Just as discussed for the wing structure, carbon fibre could perhaps be used for the landing legs and the FFU instead. Creating molds and forming sheets creates light and formable material ideal for the FFU geometry and landing legs can be made out of rods similar to the tail booms which held through all tests. As for the motor mount, it has to be attached in a better way, possibly through a hole in the FFU wall rather than on the outside the fuselage.

The future will call for more propeller tests, as the requirements make the flight speed too large for low scale drop tests.

When a more sophisticated design has been established, large scale drop tests will however be of interest.

G. Additional future improvements

After freezing the final design, no alterations to ICEBERG were allowed in order to give all I context groups a chance the enhance their data to one model. Even though the design was frozen many improvements could be made which renders ICEBERG obsolete. These improvements/considerations together with some already discussed are summarised in the following list:

- Tripping the boundary layer of the circular FFU body. This is a method to reduce the drag of non aerodynamical bodies by forcing a turbulent flow early that separates much later. This can be obtained by using dimples or other rough surfaces [3].
- Using winglets to reduce the induced drag as explained in section V-A.
- Only using carbon fiber and 3D plastic during manufacturing.
- Building landing gear for future prototypes suitable for rough terrain.
- Not using the whole ruddervator as a control surface since it renders the tail unstable. Alternatively, attaching a separate support structure.

IX. Conclusion

After using three different methods of analysis the authors have discovered two major findings. The FFU demands large wings, while the deployment requirements demand small wings. This is the primary complexity of the task. Many parameters have been iterated in an attempt to tackle the issue, but they all boil down to insufficient glide ratio and longitudinal instability. When these two problems are solved with respect to the requirements, the conceptual glider design is aerodynamically ready for the REXUS launch in 2023.

To increase the glide ratio, one can either increase the gliders lift or decrease its drag. More lift could be obtained through larger wings, but as mentioned above, that conflicts with the requirements of the task. An alternative solution would be using minor changes such as winglets or by completely changing into a less conventional design. In order to decrease the massive drag produced by the FFU a similar approach can be taken. Either using minor changes like tripping the boundary layer or by altering its shape to be more streamlined.

Simscale analysis has shown that the FFU also contributes to a substantial pitching moment which requires a larger tail to achieve longitudinal stability. In fact, ICEBERG's tail volume had to be made almost four times larger in order to counteract this contribution, making it much harder to deploy. Using a controller is another alternative to stabilise ICEBERG statically and dynamically.

The task undertaken is complex, as proven in this report, but more points in its favour rather than against it. The future calls for further investigation in CFD software and prototype testing.

Acknowledgment

The authors want to thank their supervisor Mykola Ivchenko for guidance throughout the whole thesis and providing excellent feedback. They also thank Raffaello Mariani for introducing them to the concepts of aerodynamics and discussing ideas.

Furthermore, the authors owe thanks to Victor Nan Fernández-Ayala for assistance in complex challenges during the thesis, as well as Márton Galbács for introductory courses in Computer Aided Design.

References

[1] REXUS/BEXUS. (2021, Apr) Organisations. Webpage. [Online]. Available: http://rexusbexus.net/organisations/
[2] D. P. Raymer, *Aircraft Design: A Conceptual Approach*, 6th ed. Blacksburg, VA: American Institute of Aeronautics and Astronautics, 2018.
[3] J. D. Anderson Jr., *Introduction to Flight*, 8th ed. New York, NY: McGraw-Hill Education, 2016.
[4] R. Mariani, "Basic aerodynamics," Lecture notes in Aerodynamics, Royal Institute of Technology, Stockholm, Sweden, 2021.
[5] R. C. Nelson, *Flight Stability and Automatic Control*, 2nd ed. Singapore: McGraw-Hill Companies, 1998.
[6] R. Mariani, "Performance," Lecture notes in Aerodynamics, Royal Institute of Technology, Stockholm, Sweden, 2021.
[7] ——, "Stability," Lecture notes in Aerodynamics, Royal Institute of Technology, Stockholm, Sweden, 2021.
[8] XFLR5. (2021, Jan) Xflr5. Online download. [Online]. Available: http://www.xflr5.tech/xflr5.htm
[9] Simscale. (2021, May) Simscale. Online software. [Online]. Available: https://www.simscale.com/
[10] REXUS/BEXUS. (2021, Apr) Rexus user manual. Webpage. [Online]. Available: http://rexusbexus.net/rexus/rexus-user-manual/
[11] Simscale. (2021, Apr) Tutorial: Compressible flow simulation around a wing. Online tutorial. [Online]. Available: https://www.simscale.com/docs/tutorials/tutorial-compressible-flow-simulation-around-a-wing/
[12] ——. (2021, Mar) Tutorial: Aerodynamics simulation of flow around a vehicle. Online tutorial. [Online]. Available: https://www.simscale.com/docs/tutorials/aerodynamic-simulation-vehicle/
[13] J. Tang. (2016, Apr) Figure configuration code (ieee transaction standard). Shared code. [Online]. Available: https://se.mathworks.com/matlabcentral/fileexchange/56472-figure-configuration-code-ieee-transaction-standard
[14] Simscale. (2020, Oct) K-epsilon. Online documentation. [Online]. Available: https://www.simscale.com/docs/simulation-setup/global-settings/k-epsilon/

The Deployable Wing Structure for the KTH REXUS Free Falling Unit

Jennifer Ly and Orgil Jargalsaikhan

***Abstract*—With the help of sounding rockets, the Earth's ionosphere can be studied by ejecting cylindrical units that measure various electromagnetic properties while falling. These units are also known as Free Falling Units (FFUs). The goal of this project is to turn the FFUs into autonomous gliders by designing deployable wings. A spring-loaded Scissor Structural Mechanism (SSM) was chosen as the main deploying mechanism. Furthermore, the conceptual wing design was simulated in Siemens NX and a structural analysis was performed in NASTRAN. Finally, a prototype was manufactured to confirm if the SSM would work as intended. Initial simulation results showed great promise with the proper choice of materials. Due to resource limitations, the prototype could not be compared to the simulation. Based on the prototype results, the design must be reinforced or altered to become stronger and more rigid.**

***Sammanfattning*—Med hjälp av sondraketer kan jordens jonosfär studeras genom att skicka ut cylindriska enheter som mäter diverse elektromagnetiska egenskaper medan de faller. Dessa enheter är också kända som FFUs (Free Falling Units). Målet med detta projekt var att förvandla dessa enheter till autonoma glidare genom att designa utfällbara vingar. En fjäderbelastad saxmekanism valdes som den huvudsakliga utfällningsmekanismen. Vidare simulerades den konceptuella vingdesignen i Siemens NX och strukturen analyserades i NASTRAN. Slutligen tillverkades en prototyp för att bekräfta om saxmekanismen skulle fungera som avsedd. De första simuleringsresultaten visade sig vara lovande med rätt materialval. På grund av begränsningar i resurser, kunde inte prototypen jämföras med simuleringen. Baserat på prototypresultaten måste designen förstärkas eller ändras för att bli starkare och mer styv.**

***Index Terms*—Deployment, Deployable Structure, Scissor Structural Mechanism, Glider, REXUS/BEXUS.**

Supervisors: **Nickolay Ivchenko**

TRITA number: TRITA-EECS-EX-2021:166

ABBREVIATIONS

BEXUS	Balloon EXperiments for University Students
CFRP	Carbon Fiber Reinforced Polymer
CU	Common Unit
DLR	German Aerospace Center
ESA	European Space Agency
FEA	Finite Element Analysis
FFU	Free Falling Unit
KTH	KTH Royal Institute of Technology
NACA	National Advisory Committee for Aeronautics
REXUS	Rocket EXperiments for University Students
SNSA	Swedish National Space Agency
SSM	Scissor Structural Mechanism
SU	Scissor Unit

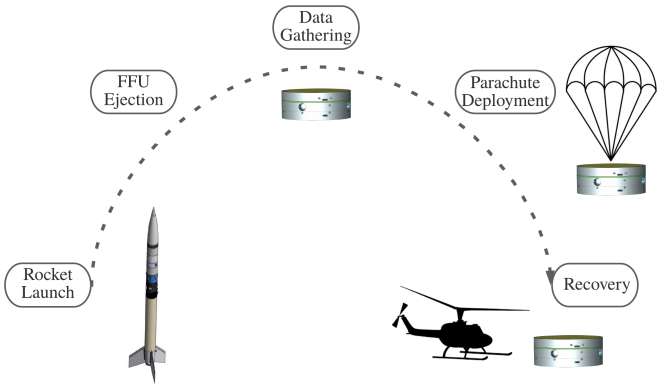


Fig. 1. A REXUS rocket launch overview. The figures are not to scale.

I. INTRODUCTION

SPACE has fascinated and allured humanity for ages. Space research has led to several important spin-off technologies, such as GPS, LED, solar cells, phone cameras, and countless other inventions that are used every day [1].

Further in the future, space research could open up opportunities such as diverting asteroids, mining precious metals, and providing another home for humanity. However, there are many things still unknown about space, even in the regions closest to Earth.

A region of space that is of special interest is the region below the hard vacuum (< 200km) and above the upper parts of the stratosphere (> 40km). This space is difficult to study due to two reasons: traditional satellites are unsuitable because of the non-negligible drag induced by the thin atmosphere, and regular weather balloons and other aircrafts cannot reach that altitude [2].

This problem necessitates another class of solution: a sounding rocket, which follows a suborbital ballistic flight path, takes measurements in the region and returns to the ground facility. By ejecting several cylindrical probes know as FFUs that record data while falling, this method can be further extended. An overview of the process is shown in Fig. 1.

Through a multilateral agreement between the DLR, SNSA, and ESA among others, sounding rockets are annually provided for students around Europe to conduct scientific and technological experiments via the REXUS/BEXUS program [3]. KTH Royal Institute of Technology has participated in the program for over a decade and is still active within this research field [4].

The goal of this project is to contribute to this program by developing the next generation of FFUs, which can au-

tonomously glide back to the launch base. This group (I2) has a specific goal of designing a deployment system for the wings and assess its mechanical performance.

The remainder of the thesis is structured as follows: in Section. II the problem formulation is defined and elaborated on. The necessary theories are outlined and developed in Section. III (Stress & Loads), Section. IV (Structures) and Section. V (Euler-Bernoulli Beam Theory). Subsequently, the method is discussed in Section. VI (Collaboration within the I-context), Section. VII (Conceptual Design), Section. VIII (Structural Analysis) and Section. IX (Manufacturing Prototype). Moreover, the results are presented in Section. X (Structural Analysis) and Section. XI (Prototype). Thereafter, the results are discussed in Section. XII and Section. XIII. Following, in Section. XIV some potential future works are proposed. Lastly, conclusions are made in Section. XV.

II. Problem Formulation

A. Aims and Objectives

In earlier KTH REXUS projects, the FFUs have landed with parachutes and been recovered by helicopter searchers.

This is a rather costly task, which is why group I1, I2, I3, and I4 - making up the I-context "Suborbital Free Flyer for Near-Earth Space Research" - strives to develop the next generation of FFUs.

Within the REXUS/BEXUS program, the I-context research team intends to develop a glider solution for the FFUs which will:

- Cut down costs associated with the retrieval of the experimental data.
- Increase the chances of the unit being found (by tracking and steering the FFU's descent back to the launch base).
- Cut down CO_2 emissions by eliminating the need for helicopter retrieval.
- Reuse material and thus reduce the climate impact.

Within this objective, I1 was responsible for the aeronautical properties of the glider solution, I2 of the deployable wing structure, I3 of the control system, and I4 of the electrical power system. More specifically, I2 had the responsibility of:

- Developing a deployment system for the wings.
- Conducting an initial structural analysis.
- Investigating manufacturability of the system.
- Building a prototype for testing.

B. Constraints

Before deployment, the wings must be stowed inside of the FFU. More precisely, the wings must fit inside the upper part of the FFU, called the Common Unit (CU), that contained a parachute earlier. The CU is a disk with a diameter of $\varnothing 240$ mm and a height of $h = 30$ mm and is displayed in Fig. 2. Provided that the CU is made of a material with a thickness of 2 mm, the wings will need to fit inside a volume with dimensions $\varnothing 236$ mm and $h = 26$ mm.

Furthermore, the rocket will spin and experience a high g-force during launch. To prevent the risk of parts flying around inside the CU, the deployment system must be robust and strong enough to survive the launch vibrations. The robustness will also be pivotal during deployment, as the wings should survive the forces caused by acceleration, vibration, and rotation in combination with the high altitude drop.

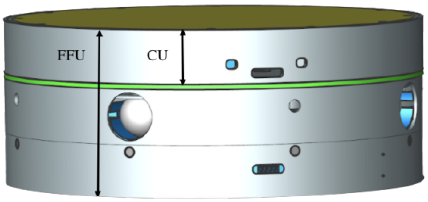


Fig. 2. The CAD design of an FFU from an earlier KTH REXUS project. The CU makes up the upper part of the FFU.

III. Theory — Stress & Loads

A. Aircraft Loads

Two primary forces act on an aircraft during flight: aerodynamic and inertial forces.

Aerodynamic forces are generated due to the interaction between the aircraft surfaces and the atmosphere. The aerodynamic forces can be divided into pressure forces (also known as lift force) and shear forces (also known as drag force). Lift is generated because of the pressure differential between the upper and lower surfaces of the aircraft. The lift force is determined by:

$$L = \frac{1}{2}\rho v^2 S C_L \tag{1}$$

where ρ is the air density, v is the true airspeed, S is the surface area and C_L is the lift coefficient. Furthermore, viscous effects create fluid resistance between the aircraft surface and the air itself, which in turn generates drag. The drag force is in the opposite direction of motion. The mathematical expression for drag is the same as for lift, only that the lift coefficient C_L is substituted for the drag coefficient C_D.

Lastly, inertial forces are generated because of the body resisting a change in its velocity. These include the gravitational force and the centripetal force when entering a turn. The gravitational force (weight *W*) will be the only inertial force considered in this work and is defined according to Newton's second law:

$$W = mg \tag{2}$$

where $g = 9.82$ m/s^2 is Earth's gravitational acceleration.

During glide, a glider will experience a lift force which is equal to or smaller than the weight [5].

B. Major Structural Stresses on an Aircraft

During flight, an aircraft can encounter four types of structural stress loads, *axial loads (tension & compression), shear loads, bending loads* and *torsion loads* [6].

Axial loads occur when a force is applied along the object's line of axis. These loads can be generated because of pressure

and thermal loads, which either pulls the object into tension or compression.

Bending loads occur when a force is applied perpendicular to the object's surface. For example, bending loads can be generated during flight as the force resultant of the lift and the weight. This force causes the upper surfaces of the wings to be compressed, while the lower surface of the wings is pulled into tension.

Shear loads arise when two adjacent planes of an object are pulled past one another in opposite directions, which can be generated from drag.

Lastly, torsion loads occur in the form of twisting as torque is applied to an object. An example is when the wings experience divergence; an aeroelastic phenomenon where the wings twists and deflects due to increasing aerodynamic forces [7].

IV. Theory—Structures

A. Aircraft Wing Structures

The wings of an aircraft are essential components that generate lift and support the aircraft during flight. Numerous wing design components, including aspect ratio, sweep, dihedral angle etc. affect flight characteristics.

Typically, the wings consist of an internal structure, covered by a skin that can act as structural reinforcement and provide aerodynamic features to the wings. The principal parts for the internal structure are longitudinal members, such as stringers and spars, which resist bending loads and support the skin against buckling. Additionally, transverse members, such as airfoil ribs, resist transverse shear loads and give the wings their shape. The structural members are displayed in Fig. 3.

The principal components can be reinforced by trusses and I-beams and work together to withstand major structural stresses [6].

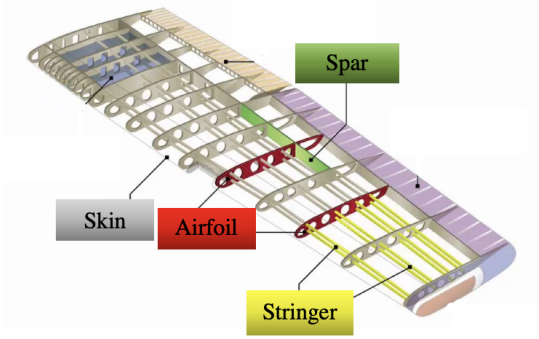


Fig. 3. The structural members of an aircraft wing, where the skin has been left out to properly show the internal structure [8].

B. Scissor Structural Mechanism

A scissor structural mechanism (SSM) consists of several, connected scissor units (SUs). By moving a single point, all the SUs deploy synchronously, enabling a rapid transformation in shape and size. Each SU consists of two bars that are crosswise interconnected at the intermediate point by a revolute joint, which allows the bars to rotate around an axis normal to the SU plane. The intermediate point in the center of the bars is called the *pivot point*, while the intermediate point between different SUs can be referred to as *hinged points*. Furthermore, imaginary lines called *unit lines* run through each SU's hinged points, which is displayed in Fig. 4. Depending on the location of these lines during the deployment and configuration in which the units are assembled, different geometric properties and kinematic behaviors can be obtained.

There are three principal types of scissor units: *translational, polar* and *angulated units*. A translational SU is characterized by parallel unit lines during deployment. By contrast, polar and angulated SUs have unit lines that intersect at one point [9].

Regardless of desired SU, various geometric constraints must be met to make the SUs geometrically compatible and the SSM fully deployable. For straight bars, which are used in this work, one principal constraint is the so-called *deployability constraint*. Considering the two SUs displayed in Fig. 4, the deployability constraint states that the sum of the semi-lengths on both sides of the unit line should be equal:

$$a_i + b_i = a_{i+1} + b_{i+1}. \tag{3}$$

Eq. 3 ensures that all SUs are capable of reaching a compact form and can be derived using the cosine rule to find distance $\overline{A_1 B_1}$ in Fig. 4:

$$a_i^2 + b_i^2 - 2a_i b_i \cos\phi_i = a_{i+1}^2 + b_{i+1}^2 - 2a_{i+1} b_{i+1} \cos\phi_{i+1}. \tag{4}$$

In compact form, the SUs are theoretically reduced to a single line. Hence, $\phi_i = \phi_{i+1} = \pi$, resulting in Eq. 3.

The SUs considered in this work are translational units. The most general translational SUs consist of two straight bars of proportional lengths. The SUs can either be regular or irregular depending on whether the intermediate point is located centrally or eccentrically (which is determined by the constant k). Furthermore, the SUs can either move in a linear 1D motion or 2D motion and are then called plane-translational or curved-translational units (determined by the constant e). The different forms of translational units are displayed in Fig. 5.

The SSM is defined by the span of the whole system (S), the span of one SU (w), the thickness of one SU (t), the number of SUs (N), the semi-length of the bars (l) and the angle between the bars (ϕ). Given that S, l and ϕ are known, the following parameters can be defined using sine and cosine rules [10]:

$$t = \sqrt{2l^2 + 2e^2 - 2(l^2 - e^2)\cos\phi} \tag{5}$$

$$w = \frac{l^2 - e^2}{t}(1 + k)\sin\phi. \tag{6}$$

The SUs of interest in this work are regular plane-translational units, meaning $k = 1$ (regular) and $e = 0$ (plane-translational). From Eq. 5 and 6 following formulas are generated:

$$t = l\sqrt{2(1 - \cos\phi)} \tag{7}$$

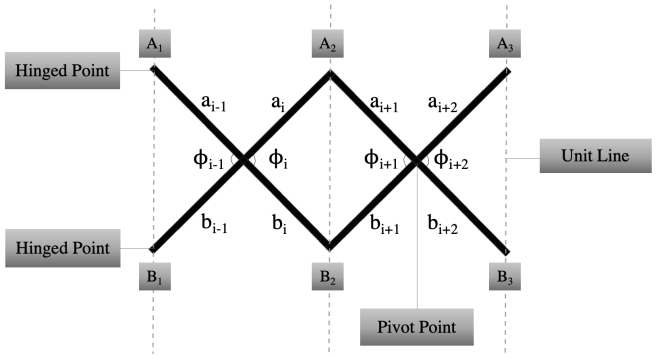


Fig. 4. Two regular translational SUs.

$$w = \frac{l^2}{t} 2 \sin \phi. \tag{8}$$

With the help of simple trigonometry, the thickness t and the span of one SU w can also be determined according to [9]:

$$t = 2\sqrt{l^2 - \frac{w^2}{4}} \tag{9}$$

$$w = 2l \cos \frac{\phi}{2}. \tag{10}$$

Eq. 9 is equivalent to Eq. 7 by substituting w in Eq. 9 by Eq. 10. The equations assume dimensionless joints, which is not the case for the true design. To accommodate for the space that the joints takes up at the hinged points, Eq. 10 can be reformulated as:

$$w = 2(l - d) \cos \frac{\phi}{2} \tag{11}$$

where d is the diameter of the joint. Lastly, the number of SUs in the SSM can be calculated according to:

$$N = \frac{S}{w}. \tag{12}$$

V. Theory — Euler-Bernoulli Beam Theory

The simplest approximation of the wing is that of a cantilever beam. Since the wings experience loads in the vertical direction and the cross-section is similar to a rectangle, standard beam theory can provide a provisional estimate for the deflection. The maximum deflection of a cantilever beam with a uniformly distributed load is defined as:

$$\delta_{max} = \frac{FL^4}{8EI_y} \tag{13}$$

where F is the uniformly distributed load, I_y is the second moment of area, E is Young's Modulus and L is the length of the beam [11]. For a rectangular beam, I_y is defined according to:

$$I_y = \frac{bh^3}{12}. \tag{14}$$

Furthermore, the von Mises stress for the beam can be calculated as:

$$\sigma_{vm} = \sqrt{(\sigma_{ax} + \sigma_b)^2 + 3\tau_{sh}^2} \tag{15}$$

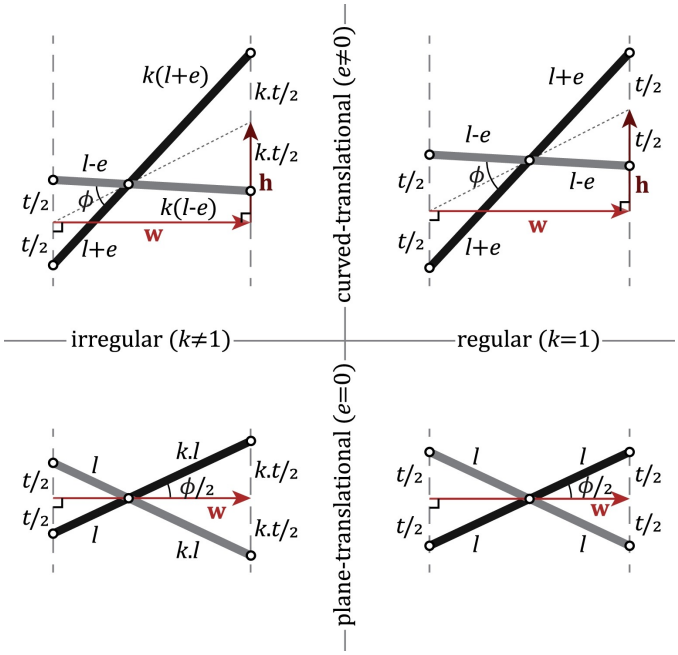


Fig. 5. Different forms of translational SUs [10].

where σ_b is the bending stress and τ_{sh} the shear stress [12], which simplifies to:

$$\sigma_{vn} = \sqrt{\left(\frac{Mc}{I_y}\right)^2 + 3\left(\frac{F}{A}\right)^2} \tag{16}$$

$$\sigma_{max} = \frac{F}{bh}\sqrt{\left(\frac{9L^2}{h^2} + 3\right)}. \tag{17}$$

Since $\frac{9L^2}{h^2} >> 3$ in this project, Eq. 17 reduces to:

$$\sigma_{max} = \frac{3FL}{bh^2}. \tag{18}$$

VI. Method—Collaboration within the I-context

Collaboration is a significant component of this project. As the entire I-context is responsible for one major objective, each group must ensure that their part of the project conforms to the rest of the REXUS-team. The project included significant collaboration with two groups in particular: I1, who is responsible for the aeronautical properties of the glider solution; and I4, who is responsible for the electrical power system.

For a successful flight, it is crucial that the wings of an aircraft are aerodynamically reliable. With the help from I1, an airfoil with proper shape, angle of incidence, and wingspan was developed. An airfoil is the cross-sectional shape of a wing and is partly defined by its chord length c, which is the distance between the leading and trailing edge. Furthermore, the angle of incidence is the fixed angle between the chord line of the airfoil and the longitudinal axis (the direction of flight). Lastly, the wingspan is the distance between the wingtips. These parameters were further refined by I2's design requirements, which include volume compatibility and structural stability. Additionally, I1's derivation of the true airspeed

and the lift coefficient was important for calculating the lift force that would later be used for the structural analysis.

Since I4 is responsible for the electrical power system of the glider solution, additional components besides the wings will take up space inside of the CU. This requires the wings to fit in such a way that accommodates I4's batteries, motors, and electrical circuits. Furthermore, I4 assisted with the placement and operation of the thermal cutter, a mechanism later discussed in Section VII.

VII. Method — Conceptual Design

The finalized design of the wings and their deployment system will consist of an internal structure, covered by a flexible but durable skin fabric. The internal structure of the wings itself will include airfoil ribs and a scissor structure. The main components inside of the CU include metal brackets, a compression spring, a thermal cutter, and a snap-fit.

In the stowed configuration, the wings will have a length of roughly 170 mm while in deployed configuration the total wingspan will be 1180 mm. The internal structure (except for the thermal cutter) of the wings in the stowed configuration and deployed configuration are displayed in Fig. 6 and Fig. 7 respectively. A more detailed picture of some of the components in the center of the CU is displayed in Fig. 9.

Furthermore, the wings will have a rectangular shape and the airfoil ribs will be placed with an angle of incidence of 8 degrees. All the components' purposes and dimensions are discussed more in detail in the subsections below.

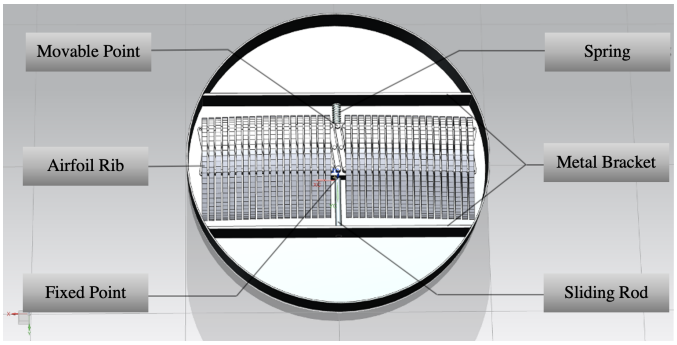


Fig. 6. Wings in the stowed configuration.

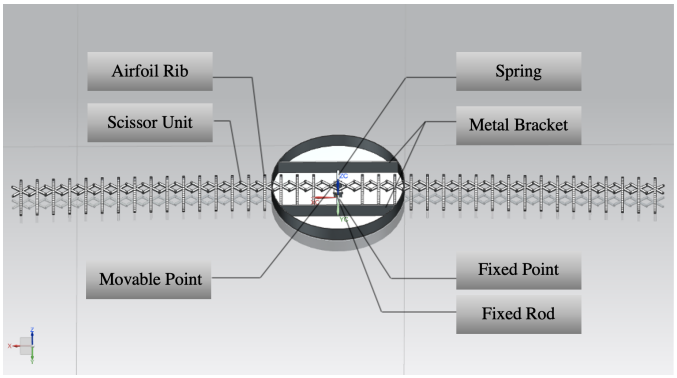


Fig. 7. Wings in the deployed configuration.

A. Airfoil Ribs

The airfoil ribs shapes the wings. Currently, there exists many types of airfoil shapes. One commonly used in aircraft wing design is the NACA airfoils, which are developed by the National Advisory Committee for Aeronautics (NACA). The airfoil shape used in this work is the NACA2415 [13], which was chosen by group I1. It has a chord length of $c = 83$ mm, a thickness of 12.45 mm, and a width of 4 mm. Additionally, each airfoil has a cavity through it, which the SSM will go through. At each side of the cavity, there are 4–6 smaller holes that the fishing line will go through to keep the airfoils parallel to each other.

B. Scissor Structure

The purpose of the SSM is to rapidly transform the wings from the stowed configuration to the deployed configuration. According to group I1's simulation, the minimum wingspan in deployed configuration should be 1140 mm. With the help of this constraint, the dimensions for the SSM were chosen or derived from Eq. 9, 11 and 12 and is displayed in Table I.

TABLE I
SSM dimensions in stowed and deployed configuration.

Symbol	Stowed configuration	Deployed configuration
S	166 mm	1180 mm
l	20 mm	20 mm
ϕ	168°	70°
w	4 mm	29.5 mm
t	40 mm	27 mm
N	40	40

The bars that make up the SUs will have a total length of 40 mm, a thickness of 4 mm and a width of 2 mm. Additionally, there will be drilled holes at the bars' pivots and hinged points. These holes will have a diameter of $\varnothing 2$ mm and are intended for the placement of the revolute joints. The idea is to have an airfoil rib mounted on the pivot point of each SU in the SSM, except for the two inner most SUs located at the center of the CU. Each SU should in turn be able to fully collapse inside of an airfoil in stowed configuration. This results in the airfoil width becoming the only factor that determines the wingspan in stowed configuration. Due to this design requirement, the dimensions of the bars were chosen to accommodate the dimensions of the airfoil ribs.

C. Metal Brackets

The purpose of the metal brackets is to stabilize the rod that the SSM is connected to. The brackets are solid metal plates that will be placed at each end of the rod. Furthermore, the brackets will serve as a fixed wall for the compression spring to push against when deploying the wings.

D. Compression Spring

The SSM is deployed by a compression spring that pushes the movable point, which is displayed in Fig. 9. This compresses the SUs and consequently expands the wings in both directions. As both wings are connected to the same movable

point, the spring has to be strong enough to deploy the whole assembly during the deployment process. The spring constant k, which determines the maximum force that can extend the wings, is approximated by:

$$E_s > W_{ext} \implies \frac{1}{2}k\Delta x^2 > \mu mg\Delta s \tag{19}$$

where E_s denotes the energy stored in the spring, W_{ext} is the work needed to deploy the entire configuration, Δx is the difference in length between the compressed and extended spring, μ is the friction constant, Δs is the difference in length between the center of mass for a folded wing and a deployed wing, and g is the Earth's gravitational acceleration.

This is equivalent to the spring storing at least the energy needed to move the wings against friction. This model is only a rough approximation of the required spring constant, since a full model of the frictional forces includes other variables as well.

Additionally, the spring must be small enough to fit inside the CU and between the wings in the stowed configuration.

E. Thermal Cutter

When the wings are in the stowed configuration, the compression spring is compressed by a fishing line. To be able to release the spring and consequently deploy the wings, the fishing line will be burned off by a thermal cutter.

The thermal cutter consists of a Kanthal wire, which is a high resistance wire that heats up when current goes through it. Furthermore, the Kanthal wire is held under a piece of plastic that can withstand high temperatures and shields the rest of the components in the CU from the heat.

Since earlier KTH REXUS projects have successfully integrated the thermal cutter in their systems [14], the same design will be used in this work.

F. Snap-fit Mechanism

Once the wings are deployed, there is a probability that the movable point will move backward. Hence, an additional mechanism that stops the wings is needed. A small and effective solution is a snap-fit [15].

The snap-fit mechanism is a mechanical lock that prevents the movable point from sliding backward and keeps the deployed wings in place. The mechanism depends on the elasticity of the material as it will bend away from the fastener. Once it reaches the required length, it will "snap" into place and prevent any movement backward. In this project, a snap-fit was designed so that it could fit beneath the SSM, while still being large enough to provide adequate support. A general snap-fit is displayed in 8. The snap-fit particularly modeled for this work is displayed in Fig. 9.

G. Skin Fabric

The skin fabric will cover the internal structure of the wings and provide a surface for the lift to be generated. The idea is to attach the fabric to each airfoil with glue, so that the airfoil ribs stay parallel to one another when deployed.

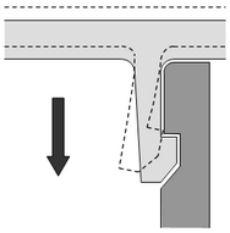


Fig. 8. A general snap-fit mechanism design [16].

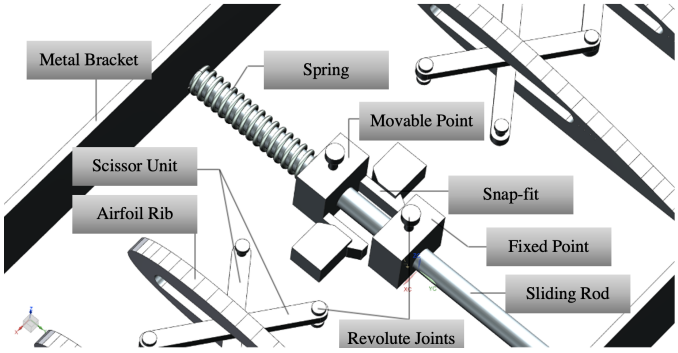


Fig. 9. A more detailed picture of some of the components in the center of the CU. Some parts of the SSM have been hidden to properly view the snap-fit.

VIII. Method — Structural Analysis

The entire assembly was modeled in the CAD software Siemens NX. The main advantage of NX is that it includes the NASTRAN solver, which is a widely used finite element analysis (FEA) program.

Since the model was created in NX, the logical decision was to perform the structural analysis in the same software. The entire assembly was assigned Aluminum 2014 as its material. Since aluminum is a type of metal, it behaves linearly until its yield point. Hence, a linear static analysis was conducted. This requires a static load assumption, which is satisfied since only constant lift was considered.

IX. Method — Manufacturing of the Prototype

A. Material

Aircraft materials must be selected with various factors in mind. An important consideration is the density of the material. As an aircraft's lift decreases with increased weight, mass must be reduced to improve the performance. This rules out heavy materials such as steel or steel alloys (7.75 to 8.05 g/cm^3) to be used for the bulk of the plane. For this reason, lighter but still strong metals such as aluminum, titanium, and magnesium are commonly used [6]. Recently, even stronger and lighter materials such as carbon fiber reinforced polymers (CFRPs) have become viable for use in aviation.

As an initial selection, Aluminum 2014 was chosen to serve as the backbone of the SSM. However, this was changed due to the student workshop constraints. If all tools were available, CFRP would be preferred over aluminum.

In regular aircraft, the backbone is spanned by metal sheets. However, this design needs a mechanism that can expand and contract while also taking up small space. This resulted in the usage of fabric to connect the airfoils and create the desired wing shape. For this purpose, several fabrics were considered, including Kevlar, nylon, and polyester, which are all light and considerably strong fabrics, often used in para- and hang gliders. In the initial model, Kevlar was considered for its strength and relative stiffness.

The bolts, nuts, and joints constitute a negligible part of the overall weight of the structure, and therefore practically any material could be used, provided it is strong enough. For this purpose, stainless steel was used due to its strength and availability.

B. Student Workshop

KTH has a student workshop intended for prototype manufacturing. There are several machines available, including a 3D printer, a laser cutter, and a CNC milling machine. Additionally, the student workshop has a range of materials available for students to use such as aluminum, acrylic plastic and plywood [17]. To get access to a certain machine, one needs an introduction.

The original idea for this work in particular was to manufacture all components in CFRP or aluminum with the help of the CNC milling machine. However, the introduction to the CNC was cancelled due to time limitations. Therefore, the prototype was decided to be built entirely in acrylic plastic. The laser cutter was used to cut out all components. Additionally, a manually controlled milling machine was used to drill holes through the airfoil ribs so that the SUs could be attached to the SSM with a pin. Components that could not be manufactured (such as the spring, the thermal cutter, and the skin fabric) were either bought or provided by the workshop.

X. Results — Structural Analysis

A. Calculations

The mass of the lower part of the FFU is approximately 2 kg. With the added components for the electronic power system (≈200g) and for the deployment system with the assumption that it is made of Aluminum 2014 (≈900g), the mass of the glider is 3.1 kg. The weight of the glider is then according to Eq. 2 approximately W = 30 N.

Since the airspeed and the air density are changing with altitude (Fig. 10 and Fig. 11 respectively), the lift will change as a function of altitude. With a surface area of $S = 0.078$ m^2 and a lift coefficient of $C_L = 0.813$, the lift force for changing airspeeds and densities from a drop of 10 km can be calculated according to Eq. 1 and is displayed in Fig. 12. According to Fig. 12, the maximum lift force the wings will experience is approximately 30 N.

Since the SSM is the main load carrier in the wing structure, a reasonable approximation is to reduce the SSM to a cantilever beam. The approximated beam will have a length of 470 mm, a thickness of 2 mm (assuming that the bars in the SUs are on the same level), and a width of 8 mm (assuming that the bars in one SU are put in parallel to each other). Young's Modulus for Aluminum 2014 is $E = 73$ GPa [18] and thus, the maximum deflection and stress can be calculated according to Eq. 13 and Eq. 18 respectively:

$$\delta_{max} = \frac{\frac{15}{0.470} \times 0.470^4}{8 \times 73 \times 10^9 \times \frac{0.008 \times 0.002^3}{12}} \approx 500\text{mm}$$

$$\sigma_{max} = \frac{3 \times 15 \times 0.470}{0.008 \times 0.002^2} \approx 660\text{MPa}.$$

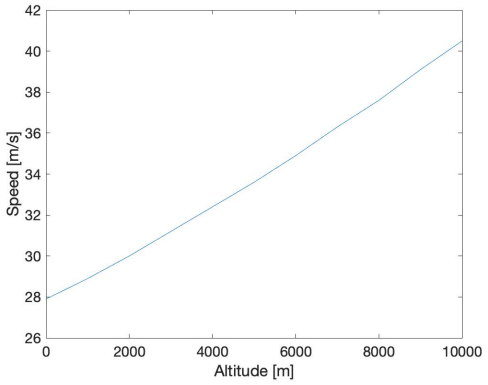


Fig. 10. The airspeed of the glider for different altitudes.

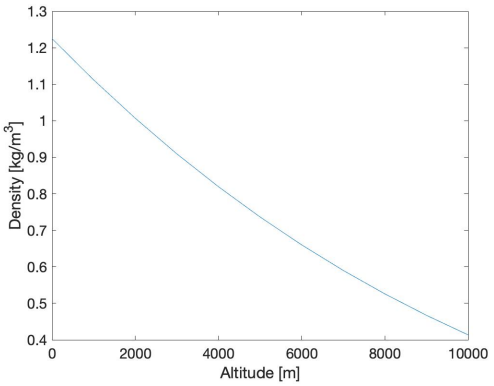


Fig. 11. The air density for different altitudes.

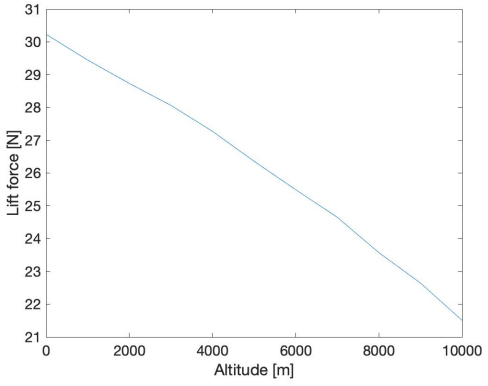


Fig. 12. The lift force for different altitudes.

B. Deflection

A structure analysis with a force of magnitude 15 N were simulated on two different models in Siemens NX:

1) A simple SSM with united SUs.
2) A complete full-scale assembly, together with the CU.

In the simpler model, the SUs were united in such a way that the revolute joints at the pivot and hinged points were nonexistent. This is not the case in reality as the bars need a pin to rotate around, which the complete full-scale assembly includes. The maximum deflections for the two cases are displayed in Table. II.

TABLE II
THE MAXIMUM DEFLECTION DERIVED IN SIEMENS NX.

Model	Maximum deflection [mm]
1	520
2	530

Furthermore, the varying deflection throughout the structure for model 1 and 2 is displayed in Fig. 14 and 15 respectively.

C. Stress

The same two models mentioned in the subsection above were considered when simulating the stress. The maximum stresses for the two cases are displayed in Table. III.

TABLE III
THE MAXIMUM STRESS DERIVED IN SIEMENS NX.

Model	Maximum stress [MPa]
1	550
2	700

Furthermore, the varying stress throughout the structure for model 1 and 2 is displayed in Fig. 16 and 17 respectively.

XI. RESULTS — PROTOTYPE

A. Design Modifications

Since each SU needs to fully collapse inside of each airfoil it is mounted on, the shape of the airfoil cavity needs to mirror the shape of the collapsed SU, including screws and nuts at the hinged points. This was not considered in the original design, which resulted in the airfoils needing to be scaled up from $c = 83$ mm to $c = 110$ mm to accommodate for the shape of the screws and nuts. This modification is displayed in Fig. 13.

Furthermore, the initial prototype was intended to be built in CFRP or aluminum due to its strength to weight ratio. Due to limitations discussed in Section. IX, the entire model was manufactured in acrylic plastic instead. Acrylic plastic is significantly weaker than any material considered, which leads to further design modifications including:

- Using larger, up-scaled airfoils to compensate for the weak points around the cavity in the original smaller airfoils (as mentioned earlier).
- Disregarding the spring mechanism, as the bars in the SUs in the center were too weak to compress without breakage, even with careful manual contraction.

Lastly, the width of the airfoil ribs changed from 4 mm in the simulated model to 5 mm in the manufactured prototype, as the 5 mm acrylic plastic was the closest available material in the workshop.

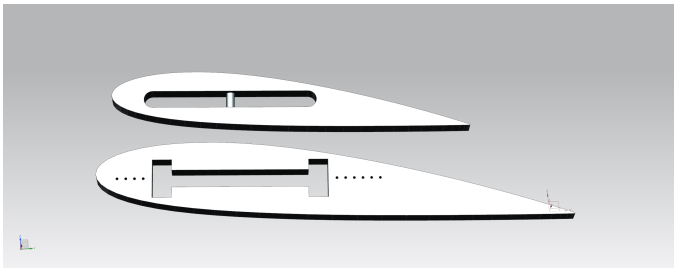


Fig. 13. The top airfoil represents the original design. The bottom airfoil represents the modified, up-scaled design used in the prototype. Although the pin (which the SU will be attached to) is only visible in the original design, a pin will also go through the modified design.

B. Finalized Prototype

Most of the components, except for the snap-fit, were manufactured and assembled. Originally, double cap screws were intended to be used at the hinged points. In the finalized prototype, however, the SUs are united at the hinged points with stainless steel M2x6 PH screws and M2 Hex Nuts. Furthermore, stainless steel pins with a diameter of $\varnothing 2$ mm and a length of 16 mm were put through the top of the airfoils and the SUs' pivot points.

Moreover, nylon fabric was bought and glued on the wings in the deployed configuration. Unfortunately, the spring and thermal cutter could not be integrated into the prototype and tested, as the SSM was too weak as explained previously in subsection above. Consequently, the metal brackets were not manufactured either, as they were determined unnecessary in this context.

The finalized prototype (without skin fabric) in the stowed and deployed configuration are displayed in Fig. 18 and Fig. 19 respectively. Furthermore, the skin interface in stowed configuration is displayed in Fig. 20.

XII. DISCUSSION—STRUCTURAL ANALYSIS

The results from the simplified model, especially when it comes to the maximum deflection, is close to the theoretical value derived from beam theory. It suggests that the beam approximation is a useful estimate of the mechanical properties of the SSM. Although the maximum stress of the second model reaches 1400 MPa, the location of that stress point could not be found in the simulation. Zooming in on Fig. 17, the simulation suggests that the maximum stress of the bars closest to the CU is approximately 700 MPa. The deviating result can be argued is an artifact resulting from the meshing of the model.

Furthermore, the full-scale model gave a larger deflection than the simplified model. This can be argued is due to the introduction of joints to the SUs. The simplified model had all the elements united together rather than multiple separate components. Intuitively, a solid body is stronger than two

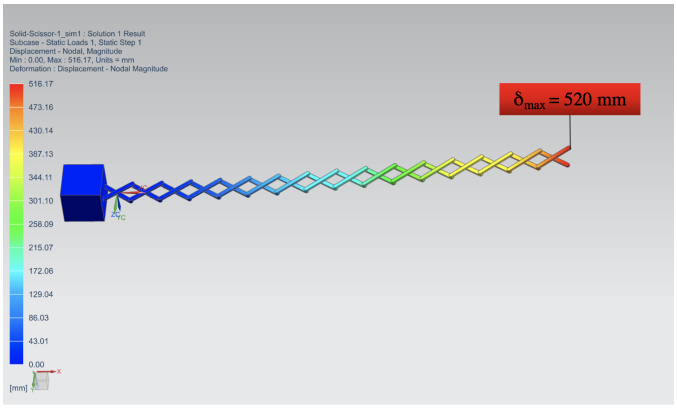


Fig. 14. The deflection of a simple SSM with united SUs for a uniformly distributed load. The maximum deflection have been marked out.

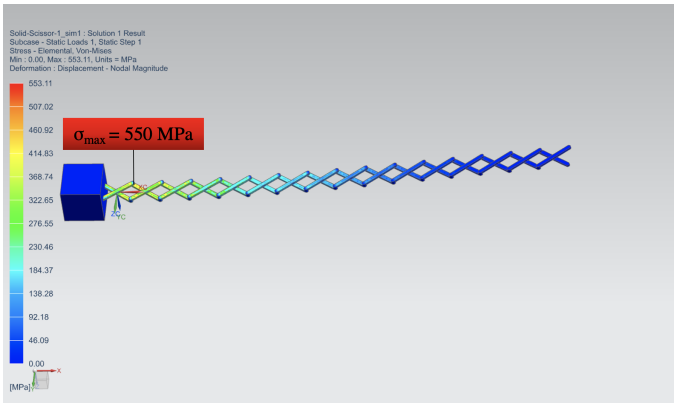


Fig. 16. The von Mises stress of a simple SSM with united SUs for a uniformly distributed load. The maximum stress have been marked out.

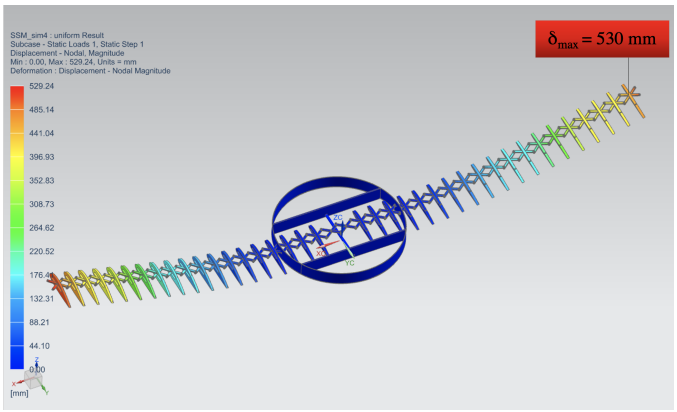


Fig. 15. The deflection of the full assembly for an uniformly distributed load. The maximum deflection have been marked out (the design is symmetrical with respect to the CU, hence the same δ_{max} applies for both wings).

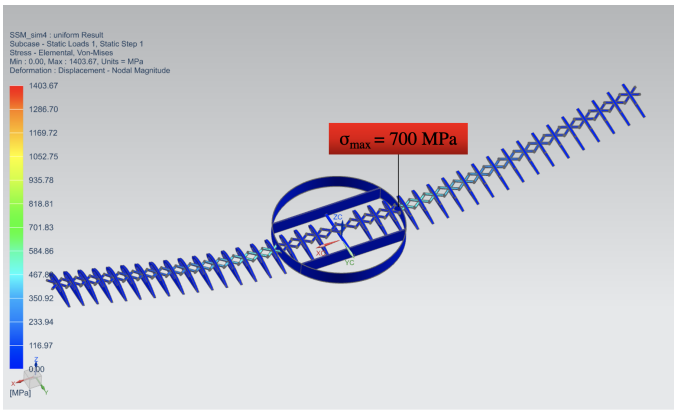


Fig. 17. The von Mises stress of the full assembly for an uniformly distributed load. The maximum stress have been marked out (the design is symmetrical with respect to the CU, hence the same σ_{max} applies for both wings).

separate components united with hinged joints. This is further supported by the simulation results.

Based on the results from the derived simulations, it suggests that there are materials strong enough to realize the scissor structural design. One of them is for example CFRP, which has a yield strength up to 3220 MPa and a density of 1.6 g/cm^3. Another potential candidate is titanium alloys, which have a yield strength up to 970 MPa and a density of 4.41 g/cm^3 [18]. These yield strengths are higher than the maximum stress the structure will experience (according to the simulations).

XIII. Discussion—Prototype

The two main purposes of the manufactured prototype were to test whether the deployment system would deploy as intended and if the wings would fit inside of the CU in the stowed configuration. With the finalized prototype, several observations were made including:

- The SSM is functional and behaves according to the simulation when the number of units is small.
- When the number of units grows, friction and disconnectedness of interior SUs with the outer SUs caused a 'lag' of deployment, i.e. the interior units achieve proper bar-angle before the outer units.
- The friction from the screws and nuts, and from the table itself, made it impossible for the wings to fully extend by moving the movable point.
- When using an elastic skin, such as nylon, the compressed fabric will want to extend itself back into the initial state. From this, one can conclude that the skin assists the SSM to deploy the wings.
- When weaker materials (e.g. acrylic plastic) are used, there is practically no stiffness or strength to the SSM as the entire structure behaves like a chain.
- Depending on the thickness of the skin material, the wings may fit inside of the CU. Subsequent simulations must account for the thickness of the skin.

XIV. Future Work

The project can be improved and further developed in several ways. First, a prototype with stronger materials for the bars can be built and tested. With stronger materials, one can test the stiffness of the SSM and determine whether it is true to the results of the structure analysis. Additionally, stronger material enables other mechanisms to be tested, such as the spring deployment.

Second, the dimensions of the bars and airfoils could be changed to optimize the structural integrity of the SSM. In

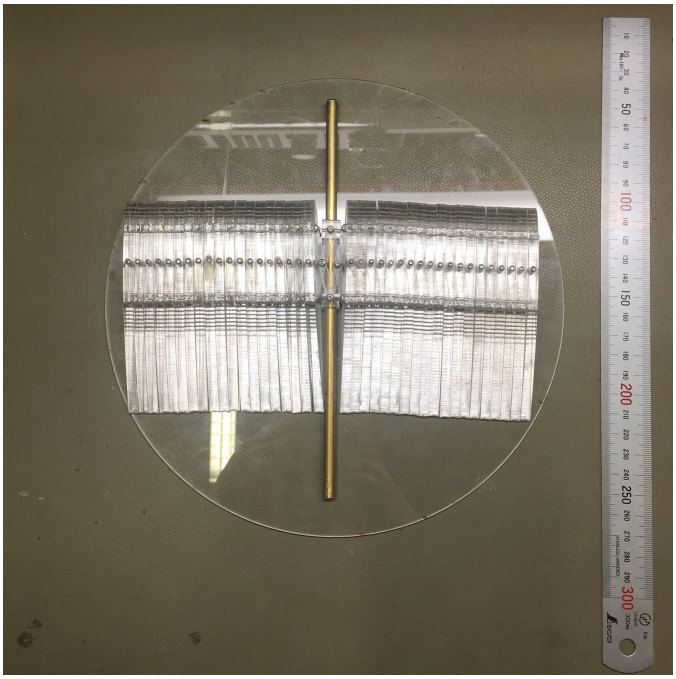


Fig. 18. The finalized prototype (without skin fabric) in the stowed configuration. Since the width of the airfoil ribs increased from 4 mm to 5 mm, an additional 1 mm was added for each airfoil. Thus, a total of 20 mm were added for each wing.

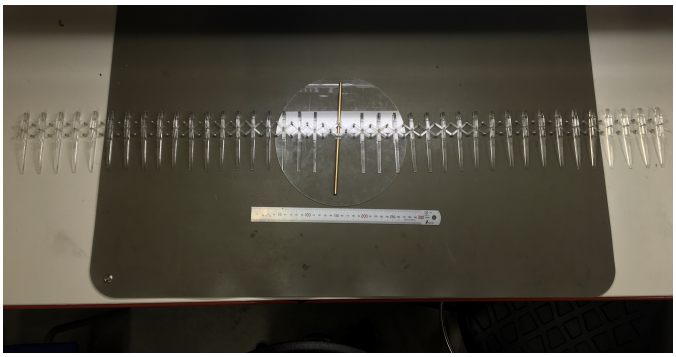


Fig. 19. The finalized prototype (without skin fabric) in the deployed configuration.

particular, the deflection of a beam-like structure is proportional to $1/h^3$, while the von Mises stress is proportional to $1/h^2$. By doubling the height, the deflection could be reduced by a factor of 8 and stress by a factor of 4. However, the airfoil dimensions must increase to achieve the reduction. The dimensions of the airfoils affect aerodynamic properties and hence it is questionable whether it can be easily changed. Moreover, the size of the bars cannot be increased arbitrarily as the size constraint of the CU limits the available volume that the internal structure can occupy.

When it comes to the wing design, the rigidity of the structure is an important property that needs to be further developed. One idea for design improvement is by introducing stringers, a rod-like structural element that goes through the airfoils and provides additional support. These stringers could be deployed through a telescopic expansion mechanism, such

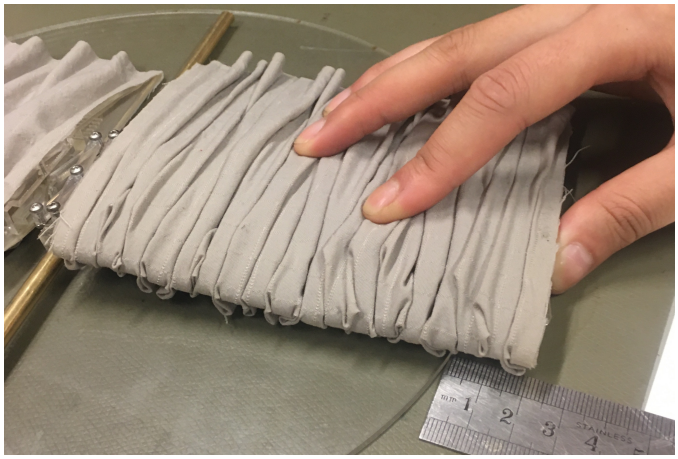


Fig. 20. The skin-interface in the stowed configuration. Since the skin tends to crease inwards as it is compressed, an additional length of 20 mm was added for each wing.

that when the SSM starts to deploy, the stringers will follow along the airfoil ribs.

Another idea is to remove the airfoil ribs entirely and instead change the shape of the bars in the SUs such that in the deployed configuration, the SUs would deploy in the form of an airfoil. By replacing small and thin bars with thicker and taller units, the structural performance of the SSM can be improved. However, there are some challenges with this concept. One of them includes the shape transformation during deployment, i.e. the initial form will not be the same as the final form. This is not a problem in the original design, as the airfoil shape is constant during the deployment and the SSM can collapse inside of the airfoils. Another problem that comes with this design is solving how to connect the SUs at the pivot and hinged points.

Moreover, the skin attachment could be altered to reduce the risk of the wings not fitting inside of the CU in the stowed configuration. One possible solution is to separate the fabric from the wings and let the SSM expand into it and stretch the fabric during deployment, like a sock.

Furthermore, a more detailed structural analysis could be performed. Specifically, an aeroelastic and vibroacoustic analysis are pivotal to ensure a successful flight. The aeroelastic analysis will investigate the interaction between the air and the wings and consider all the forces that affect the glider during flight. The vibroacoustic analysis will study the frequency response of the wings. The glider will experience vibrations during launch and flight, and this must be analyzed to guarantee a stable and robust structure.

Ultimately, the dimensions of the CU itself could be altered. By increasing the relevant dimensions of the CU, the SSM could be enlarged and thus provide a more stable structure. However, this is unlikely as the shape of the CU is limited by the size and the shape of the sounding rocket.

XV. Conclusion

The prototype and simulations show that the scissor structure is a viable design for a deployment system. The primary

challenges for this design are the strength of the material and finding a proper shape and size for the components that balances aerodynamics and structural strength. The structure can be strengthened by using stronger materials such as CFRP or titanium.

Furthermore, as discussed in the results (Section. X), the Euler-Bernoulli beam theory calculations agree relatively well with the results from the full-scale simulation. This suggests that the simplified model is an accurate estimate of the true model.

According to both the simulations and prototype, the wings can fit inside of the CU with proper choice of skin material, the width of the airfoils, and the number of SUs. The simulations indicate that the wings are capable of being extended to the desired length. This could not be confirmed with the prototype.

Furthermore, it is possible to control the length by changing the interior angles of the scissor units. However, friction between the components greatly affects the deployment.

Finally, the skin was able to properly envelop the airfoil ribs and create a shape similar to the desired airfoil shape throughout the wing.

ACKNOWLEDGMENT

The authors want to thank our supervisor Dr. Nickolay Ivchenko, for his tremendous support and valuable insights during the project. Additionally, we would like to thank Gunnar Tibert for the information about the SSM. Lastly, we would like to thank Victor Nan Fernández-Ayala for helping us with the aerodynamics and Márton Galbács for his assistance with manufacturing.

REFERENCES

[1] A. Spadoni. (2020, Apr) How technology from the space race changed the world. [Online]. Available: https://now.northropgrumman.com/how-technology-from-the-space-race-changed-the-world/

[2] G. Seibert, *The history of sounding rockets and their contribution to European space research*. Noordwijk, the Netherlands: ESA Publications Division, 2006.

[3] (2021, Apr) The REXUS/BEXUS Programme. [Online]. Available: http://rexusbexus.net/

[4] J. Jazayeri. (2020, Feb) REXUS. [Online]. Available: https://www.kth.se/sci/centra/rymdcenter/kurser-och-utbildnin/rexus-1.809393

[5] J. Anderson, *Introduction to flight*. New York, NY: McGraw-Hill Education, 2016.

[6] (2021, Apr) Introduction to Aircraft Structures. [Online]. Available: https://aerotoolbox.com/intro-airframe-structure/#How_are_Loads_Generated

[7] D. H. Hodges and A. Pierce, *Structural Dynamics and Aeroelasticity*. England: Cambridge, 2002.

[8] (2012, Aug) Aircraft structures. [Online]. Available: https://aerospaceengineeringblog.com/aircraft-structures/

[9] F. Maden, K. Korkmaz, and Y. Akgün, "A review of planar scissor structural mechanisms: Geometric principles and design methods," *Architectural Science Review (Architect Sci Rev)*, vol. 54, pp. 246–257, Aug 2011.

[10] K. Roovers and N. De Temmerman, "Deployable scissor grids consisting of translational units," *International Journal of Solids and Structures*, vol. 121, pp. 45–61, Aug 2017.

[11] (2021, May) Beam deflection tables. [Online]. Available: https://mechanicalc.com/reference/beam-deflection-tables

[12] (2021, May) Beam analysis - validation. [Online]. Available: https://mechanicalc.com/calculators/beam-analysis/validation

[13] (2021, May) Naca 2415 - naca 2415 airfoil. [Online]. Available: http://airfoiltools.com/airfoil/details?airfoil=n2415-il

[14] M. Axelsson *et al.*, "RX30 B2D2 SED v 3.1," *KTH Royal Institue of Technology Stockholm, Student Experiment Documentation*, Jan 2021.

[15] D. Evans. (2015, Feb) How to Design Snap Fit Components . [Online]. Available: https://www.fictiv.com/articles/how-to-design-snap-fit-components

[16] (2021, Apr) Design for manufacturing assembly (dfma) tips. [Online]. Available: https://mae.ufl.edu/designlab/DFMA%20Tips/DFMA%20Tips.htm

[17] (2020, Mar) Student Workshop. [Online]. Available: www.kth.se/ee/spp/education/student-workshop/studentverkstan-1.683473

[18] (2021, Apr) Matweb: Online materials information resource. [Online]. Available: http://www.matweb.com/index.aspx

Simulation and Control System Design for Autonomous Gliding to a Given Location

Mathias Schmekel and Ludvig Ringaby

***Abstract*—The aim of this project was to design a flight control system with the purpose of safely guiding a glider toward a given GPS location over a distance of at least 50 km. More specifically, the aim was to develop a control system for autonomous gliding and implement it on a given data hub containing sensors, GPS-module, microcontroller and a Field Programmable Gate Array (FPGA). A SIMULINK simulation environment has been developed for simulating flight dynamics and the digital implementation of some of the on-board hardware. The simulation environment also serves as a platform to tune controllers and implement most of the necessary logic for the control system. For the reference heading, a trigonometric formula is used along with latitude/longitude coordinates to calculate the turn-angle necessary to travel along the shortest path between two points. Four negative feedback control loops are used to track the reference heading and achieve maximum glide ratio. The project has been conducted with mixed success, where the implementation part of the project has suffered great drawbacks mainly due to problems in developing the simulation environment. In spite of this, the open loop simulation outputs promising results where the glider behaves as expected and is considered realistic enough to be a suitable environment in which to develop a flight control system. In addition, given that the gliders geometry offers reasonable aerodynamic stability, it is shown in this thesis that the proposed control system architecture and heading reference system is sufficient to steer the glider to the given location under calm atmospheric conditions.**

***Sammanfattning*—Målet med detta projekt var att utveckla ett styrsystem för att på ett säkert sätt styra ett segelflygplan mot en given GPS-position över ett avstånd på minst 50 km. Mer specifikt var målet att utveckla ett styrsystem för autonom glidning och implementera det på en given datahubb som innehåller sensorer, GPS-modul, mikrokontroller och programmerbar logik (FPGA). En SIMULINK-simuleringsmiljö har utvecklats för att simulera flygdynamiken och för digital implementering av några av de givna hårdvarukomponenterna. Simuleringsmiljön agerar också som en plattform för att justera regulatorer och implementera det mesta av den nödvändiga logiken för styrsystemet. För referensriktning används en trigonometrisk formel tillsammans med latitud/longitud koordinater för att beräkna den sväng-vinkel som krävs för att färdas längs den kortaste vägen mellan två punkter. Fyra regulatorer används till att följa rätt kompassriktining samt maximera flygtid. Projektet har genomförts med blandad framgång, där genomförandet av projektet har blivit lidande främst på grund av problem med att utveckla simuleringsmiljön. Trots detta ger simulersmiljön lovande resultat där segelflygplanet beter sig som förväntat och anses därmed vara en realistisk nog plattform för att utveckla ett kontrollsystem i. Dessutom, givet att geometrin av segelflygplanet ger rimlig aerodynamisk stabilitet, framgår det i denna rapport att den föreslagna styrsystemarkitekturen och referensriktningslogiken är tillräcklig för att styra segelflygplanet till den givna positionen under lugna atmosfäriska förhållanden.**

***Index Terms*—SIMULINK, autopilot, control, simulation, glider**

Supervisor: Nickolay Ivchenko

TRITA number: TRITA-EECS-EX-2021:167

I. Introduction

Society as we know it relies heavily on the use of space systems, which provide the basis for human communication and navigation. To know the conditions in which these systems operate, and to include the effects of space phenomena on the global atmospheric system, one must understand the near-earth space region. Due to satellite orbits not being stable below 300 km altitude, the lower part of the ionosphere is not as extensively studied as the space higher up. This has made studies of the region between 50 km and 200 km territory of the *sounding rockets'* - suborbital probes that return to earth after a relatively short trajectory. Since the 1950s, sounding rockets carrying payloads in the form of experiments or measuring devices gained interest since they give an opportunity to study the near-earth space.

A possible configuration is to have the payloads consist of multiple units carried by the same rocket that may be ejected at a certain altitude, if desired. Once ejected, the now free falling unit (FFU) will take measurements and store data of interest as it plunges toward the earth. Because of the sounding rocket's trajectory, the FFU will land far away from the launch site and need to be retrieved by helicopter. Retrieving a FFU by helicopter works in practice, but is costly and can be time consuming. The desire to instead have them autonomously fly back to the launch site birthed four BSc projects focused on different aspects of the FFUs recovery unit (RU).

The purpose of this BSc project is to develop an autonomous flight control system and implement it on the software of the embedded computer system previously used in the sounding rocket experiments at KTH Royal Institute of Technology. Previous years the FFUs have carried experiments with their recovery units developed by the REXUS team at KTH and as such this project is heavily based on the inherited hardware and software from those projects, more specifically the projects PRIME and B2D2 [1], [2].

II. Glider specifications

A. Glider geometry

This section describes the current iteration of glider design at the time of writing this thesis. The glider consist of a large

cylindrical disk (the FFU), wings and an inverted V-tail, see Fig. 1. Although not visible in Fig. 1, the inverted V-tail has control surfaces which can be turned using actuators and motors. Turning the control surfaces offers the only way to change the glider orientation and direction from a control perspective, and because they are mounted at an angle relative the glider body they serve as both an elevator and a rudder. Elevator and rudder are the control surfaces one would find on a conventional (inverted T) tail that turn up/down or right/left respectively (as seen from behind). The disk is internally divided into three different sections where the top section is the Common Unit (CU), the middle section is the Boom Deployment Unit (BDU) and the lowest section the Base Unit (BU). The BDU and BU are responsible for deploying measuring devices and collecting data while the CU contains and deploys the wings and tail and stores the hardware required to steer and control the glider. This disk is mounted in a Rocket Mounted Unit (RMU) that is part of the sounding rocket that launches the disk into space.

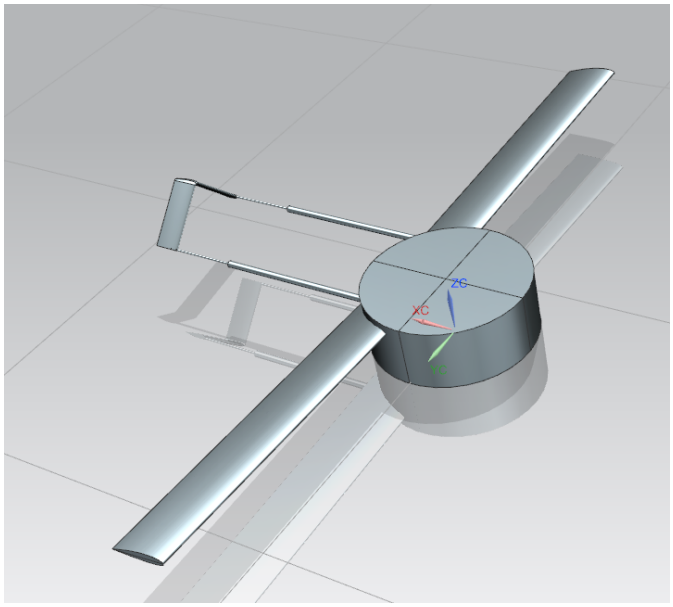


Fig. 1. The glider design used in this BSc project.

The specific glider parameters of this project are given in table I set by BSc project I_1. For more information about the design choice see BSc project I_1:s thesis [3].

TABLE I
DESIGN PARAMETERS

Wingspan b = 0.900 m	Mean chord length c = 0.083 m	Wing surface area S = 0.075 m^2
Mass m = 3.004 kg	Incidence angle = 7.5°	Dihedral angle = 2°

B. On-board hardware and software

The on-board hardware is heavily based on previous REXUS projects. The hardware presented in this section is only that of primary importance from a control perspective and therefore some components are not mentioned, for those interested in the full documentation of the hardware see [4].

The hardware of primary importance for this project is a ProASIC3 Field Programmable Gate Array (FPGA) (A3P250-VQG100I), an ARM Cortex-M7 based microcontroller (STM32F767VIT6), a SD card, a MEMS gyroscope (L3GD20H), an E-compass (LSM303AGR) and the GPS (MAX-M8). Furthermore, BSc project I_4 has developed additional components required to implement the control system. For more information about the additional components see [5].

The FPGA handles the communication with the sensors and GPS. The sensor and GPS data is received, packaged and sent to the microcontroller via two separate Universal Asynchronous Receiver Transmitter (UART) lines, where the GPS UART line has a baud rate of 9600 bit/s and the UART line used for sensor data has a baud rate of 115200 bit/s. Due to the low level of hardware in the FPGA it is programmed in VHDL, a hardware description language standardized by IEEE. The microcontroller on the other hand is programmed in C code. Furthermore, the microcontroller is equipped with a Floating Point Unit (FPU) to increase the overall speed and make it more robust.

III. THEORY

A. Reference frames

This section introduces the different reference frames used in this project.

1) The body-fixed reference frame F_B: The body-fixed reference frame is an orthogonal right handed coordinate system with its origin fixed in the gliders center of gravity (c.g.). The x_B axis points through the gliders nose, the z_B axis is perpendicular to x_B and points down thus making the x_Bz_B-plane the plane of symmetry of the aircraft. The y_B is perpendicular to the plane of symmetry and points to the right to satisfy the right hand rule. See Fig. 2 - 3.

2) The vehicle-carried NED reference frame F_V: The vehicle-carried North East Down reference frame is an orthogonal right handed coordinate system with its origin fixed in the gliders c.g. Here, the x_V axis is aligned with the local magnetic field lines horizontal component, the y_V axis is perpendicular to the x_V axis and points towards magnetic east with the same magnetic declination as for the north direction and the z_V axis is aligned with the force of gravity. Here and throughout this thesis, the term 'magnetic north' is used to refer to the local horizontal direction of the earths magnetic field lines. See Fig. 2 - 4.

3) The wind axis reference frame F_W: The wind axis reference frame is a right handed coordinate system with its origin fixed in the gliders c.g. The x_W axis points in the direction of the velocity vector of the glider, the z_W axis points

in the opposite direction of the force of lift and the y_W axis points to the right of x_W as seen looking in the direction of travel, such that it satisfies the right hand rule. See Fig. 2 - 3.

4) The earth-fixed NED reference frame F_E: The earth-fixed North East Down reference frame is an orthogonal right handed coordinate system with its origin at an arbitrary point on the surface of the earth. The x_E axis points in the polar direction due true north, the y_E axis points east, and the z_E axis is aligned with the force of gravity to satisfy the right hand rule. This reference frame is only used for simulation purposes to place the glider approximately 70 km from Esrange Space Center at the start of the simulation and to generate latitude longitude coordinates. See Fig. 4.

5) The LLA reference frame: The Latitude Longitude Altitude reference frame (also known as a geodetic reference frame) is used to determine the gliders position in space and the desired heading needed to get from the current position to the destination. In the simulation environment this is coupled with the F_E reference frame in order to simulate a trajectory in the north of Sweden.

6) The XFLR5 reference frame F_C: XFLR5 is a software used to design and analyse the performance of wings, airfoils and planes [6]. The XFLR5 reference frame is a right handed coordinate system with its origin close to the vehicles center of gravity. The x_C and z_C axes point in the negative direction of their counterpart in the wind axis reference frame. The y_C axis is parallel to the y_B axis.

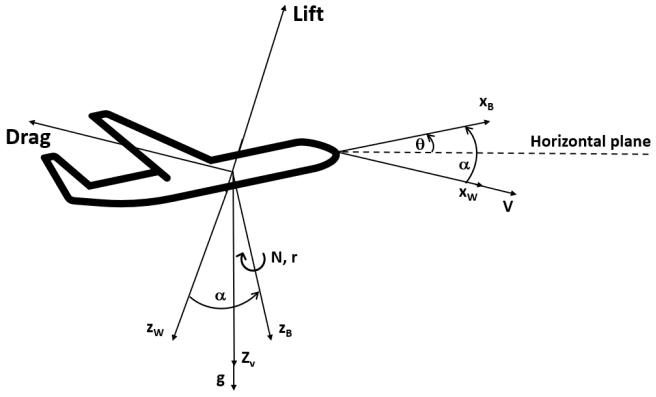


Fig. 2. Shows the x and z axes directions of the F_V, F_B and F_W reference frames and the velocity V. Even though not shown in the figure, the $x_V y_V$ plane is parallel to the horizontal plane. In addition, the figure shows the direction of the force of lift, drag, gravity (denoted by g). It also shows part of the relationship between the reference frames via the angle of attack α and pitch angle θ. The yaw rate and yaw moment r and N and their positive direction are also shown. The y axis of all reference frames have been omitted.

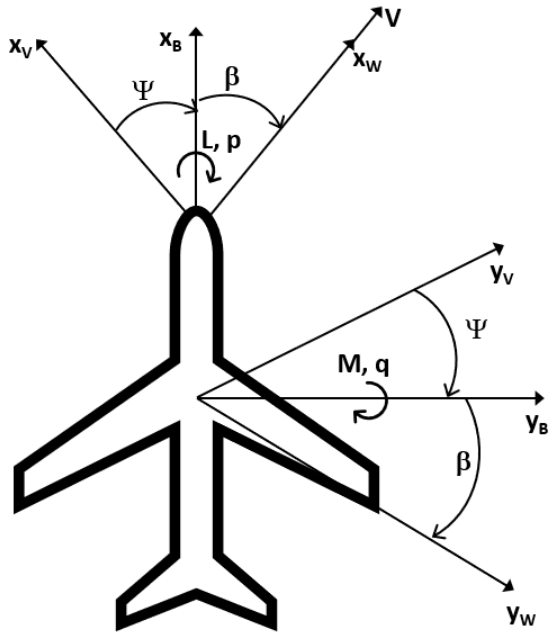


Fig. 3. Shows the x and y axes directions of the F_V, F_B and F_W reference frames and the velocity V. In addition, the figure shows part of the relationship between the reference frames via the yaw angle Ψ and side slip angle β. It also shows the positive directions for the roll rate p, rolling moment L, pitch rate q and pitching moment M. The z axis of all reference frames have been omitted.

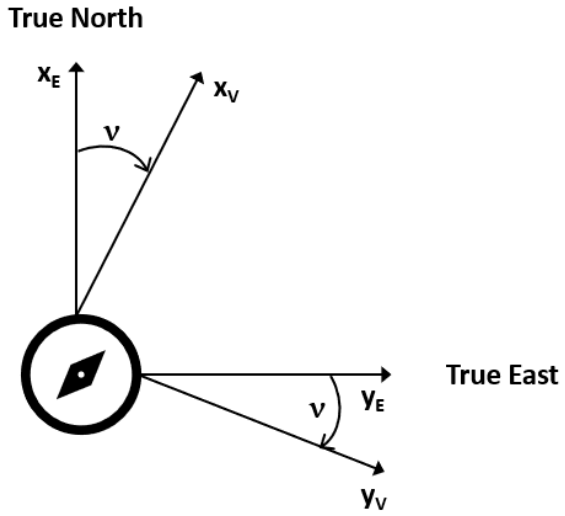


Fig. 4. Shows the x and y axes directions of the F_E and F_V reference frame, and the relationship between them via the local magnetic declination ν.

B. Relationship between reference frames

The inertial F_E and vehicle-carried F_V NED reference frames are very similar but differ in two ways:

- The origin of F_V is at the vehicle c.g. and the origin of F_E can be chosen arbitrarily on the surface of the earth.

- The north and east axis of the reference frames are offset by the local magnetic declination.

Magnetic declination is a result of magnetic field lines following curved paths toward north. The magnetic declination offset means that the x_V and y_V axes are rotated an angle ν clockwise (as in due east) from the x_E and y_E axes, see Fig. 4. The angle ν denotes the local magnetic declination which is assumed to be constant during the entire flight. It is further assumed that during flight the distance between the origins of F_E and F_V will never be so large that their local direction of gravity differs, meaning z_E and z_V are always aligned.

The body-fixed F_B and vehicle-carried F_V reference frames are similar only in the fact that they share the same origin, and the relationship between them can partially be seen in Fig. 2 - 3. The difference between the two is the inclination of the unit base vectors which can be described using angles. Since the x_V and y_V axes span the horizontal plane and z_V is aligned with the force of gravity, the angles that describe this inclination define the vehicles so called *attitude* with respect to the earth. These angles are defined in this thesis as the yaw angle Ψ, the angle between magnetic north and the projection of the x_B axis onto the horizontal plane, pitch angle θ, the angle between the x_B axis and the horizontal plane, and roll angle φ, the angle between the y_B axis and the horizontal plane. Strictly speaking, φ is the angle between the y_B axis and the horizontal plane in rotation about the x_B axis if $\theta = 0$. Note here that in other literature regarding aerodynamics, it is more common to denote φ as the bank angle. To summarize, the angles Ψ, θ, φ define the vehicles *attitude* which is information about the vehicles orientation and heading with respect to the surface of the earth.

To go from the F_V reference frame into the F_B reference frame, three separate rotations must be performed and the order in which they are performed matters since it affects the final position. This becomes more clear when considering the three individual rotation matrices T_φ, T_θ and T_Ψ and the fact that matrices are not commutative. The so called aerodynamic sequence is the rotation order Ψ, θ and φ and the order that this thesis will use exclusively. This rotation order then describes an initial rotation due east by Ψ about the z_V axis in the horizontal plane, a second rotation by θ upward from this intermediate position and a final rotation by φ around the x_B axis. Multiplying their respective transformation matrices together in a way that applies the transformations according to the aerodynamic sequence, the rotation matrix from the F_V reference frame into the F_B reference frame T_{VB} is obtained as

$$T_{VB} = T_\varphi T_\theta T_\Psi = [T_{VB1}, T_{VB2}, T_{VB3}] \quad ,$$

where T_{VB1}, T_{VB2} and T_{VB3} are the columns of T_{VB}

$$T_{VB1} = \begin{bmatrix} \cos\Psi\cos\theta \\ \cos\Psi\sin\theta\sin\varphi - \sin\psi\cos\varphi \\ \cos\Psi\sin\theta\cos\varphi + \sin\psi\sin\varphi \end{bmatrix} \tag{1}$$

$$T_{VB2} = \begin{bmatrix} \sin\Psi\cos\theta \\ \sin\psi\sin\theta\sin\varphi + \cos\psi\cos\varphi \\ \sin\psi\sin\theta\cos\varphi - \cos\psi\sin\varphi \end{bmatrix} \tag{2}$$

$$T_{VB3} = \begin{bmatrix} -\sin\theta \\ \cos\theta\sin\varphi \\ \cos\theta\cos\varphi \end{bmatrix} \quad .$$

The rotation matrix from the earth-fixed frame F_E into the F_B body frame, T_{EB}, is then obtained by exchanging Ψ in equations (1) and (2) with

$$\Psi_E = \Psi - \nu \quad ,$$

where Ψ_E is the angle between true north and the projection of the x_B axis onto the horizontal plane and the third columns remains the same i.e. T_{VB3} = T_{EB3} [7].

The body reference frame F_B and the wind axis reference frame F_W share their origins at the c.g. of the vehicle but differ in their orientation of the unit base vectors, see Fig. 3 - 2. The angles used to describe this difference in inclination are commonly referred to as the side slip angle β and the angle of attack α. The side slip angle β is the angle between the x_B axis and the projection of the velocity vector onto the $x_B y_B$ plane, and the angle of attack α is the angle between the x_B axis and the projection of the velocity vector on the $x_B z_B$ plane. Once again, to go from F_B to F_W the order of rotation matters and has to be performed as rotating an angle β then α.

C. Aerodynamics

The aerodynamics of an aircraft describe how the body of the aircraft interacts with the air surrounding it. Anything that moves through the air is affected by it. With that said, using aerodynamics one can explain how an aircraft can fly through the air.

1) Equations of motion: In order to control a system, it is vital to understand the system dynamics. This section presents equations often used to model such dynamics under certain flight conditions.

Knowing the glider is symmetric and assuming a calm atmosphere, the forces acting on the glider in the body reference frame F_B can be described according to [7] as

$$m\dot{u} = X - mg\sin\theta + m(rv - qw) \tag{3}$$

$$m\dot{v} = Y + mg\cos\theta\sin\varphi + m(pw - ru) \tag{4}$$

$$m\dot{w} = Z + mg\cos\theta\cos\varphi + m(qu - pv) \quad . \tag{5}$$

Here u, v, w are the instantaneous components of linear velocity in the x_B, y_B, z_B directions and the dot convention is used here and throughout this paper to denote time derivatives such that $[m\dot{u}, m\dot{v}, m\dot{w}]^T$ represent the net forces acting on

the glider according to Newtons second law. The first terms X, Y and Z denotes the aerodynamic forces of drag, side force and lift acting on the glider, and the second term in each equation is the force of gravity, taking into account the body frames rotation in pitching angle θ and rolling angle φ with respect to the vehicle-carried reference frame F_V. Lastly, the final terms in the force equations relate to gyroscopic motion and describes the tendency of a rotating object to maintain its rotation. Here p, q and r (commonly referred to as roll, pitch and yaw rate) denote the instantaneous angular velocities about the x_B, y_B, z_B axes in the F_B body reference frame respectively. The components of aerodynamic force are obtained as

$$X = C_D Q S \tag{6}$$

$$Y = C_Y Q S \tag{7}$$

$$Z = C_L Q S \quad , \tag{8}$$

where C_D, C_Y and C_L are the dimensionless coefficients of drag/friction, side-force and lift respectively. S is the reference wing area and Q denotes the dynamic pressure as a function of air density ρ and the gliders velocity V

$$Q = \frac{1}{2}\rho V^2 \quad . \tag{9}$$

In addition, the dimensionless coefficients are used to add contributions in forces and moments from several factors. Adding contributions due to the gliders angular rates is done with the help of the dimensionless roll, pitch and yaw rates $\hat{p}, \hat{q}, \hat{r}$,

$$\hat{p} = \frac{pb}{2V} \tag{10}$$

$$\hat{q} = \frac{qc}{2V} \tag{11}$$

$$\hat{r} = \frac{rb}{2V} \tag{12}$$

and the so called stability derivatives [7]. In equations (10) - (12) the rates have been multiplied with $\frac{b}{2V}$ or $\frac{c}{2V}$, effectively making the rates dimensionless since

$$[\hat{p}] = [\hat{q}] = [\hat{r}] = \left[\frac{pb}{2V}\right] = \left[\frac{qc}{2V}\right] = \left[\frac{rb}{2V}\right] = \tag{13}$$

$$= \frac{\text{rad}}{\text{s}}\frac{\text{m}}{\text{m/s}} = [1] \quad .$$

In equation (13) the notation $[x]$ is used to denote the unit of x and [1] means unitless. Here b and c correspond to the glider reference wingspan and mean chord length respectively, where chord is the imaginary straight line joining the leading and trailing edge of an aerofoil. The stability derivatives due to angular rates are the dimensionless coefficients differentiated with respect to the dimensionless angular rates $\hat{p}, \hat{q}, \hat{r}$ such that

$$C_{L_q} = \frac{\partial C_L}{\partial \hat{q}} \tag{14}$$

$$C_{Y_p} = \frac{\partial C_L}{\partial \hat{p}} \tag{15}$$

$$C_{Y_r} = \frac{\partial C_L}{\partial \hat{r}} \quad . \tag{16}$$

Using expressions (10) - (12) with (14) - (16) the dimensionless side-force and lift equations can be expanded to

$$C_Y = C_Y(\beta, \delta_r) + C_{Y_p}\hat{p} + C_{Y_r}\hat{r} \tag{17}$$

$$C_L = C_L(\alpha, \delta_e) + C_{L_q}\hat{q} \quad , \tag{18}$$

where the first terms are the contributions due to side slip angle β, angle of attack α and the effect due to rudder and elevator deflections δ_r and δ_e respectively. The rest of the terms are contributions due to the gliders own angular rates which depending on the sign of the stability derivatives and angular rates either amplifies or dampens the force.

The equations describing the net moments about the x_B, y_B, z_B body axes can be formulated simply as

$$I_{xx}\dot{p} = L + (I_{yy} - I_{zz})qr + I_{xz}(pq + \dot{r}) \tag{19}$$

$$I_{yy}\dot{q} = M + (I_{zz} - I_{xx})pr + I_{xz}(p^2 - r^2) \tag{20}$$

$$I_{zz}\dot{r} = N + (I_{xx} - I_{yy})pq + I_{xz}(qr - \dot{p}), \tag{21}$$

under certain assumptions [7]. Like in the case of the forces, calm atmospheric conditions are assumed as well as a symmetric glider. In particular, since the y_B axis is perpendicular to the gliders plane of symmetry (the $x_B z_B$ plane) the products of inertia $I_{xy} = I_{yz} = 0$ and placing the origin of the body reference frame F_B in the gliders center of gravity is what yields the simplified equations (19) - (21). Here I_{xx}, I_{yy}, I_{zz} denote the moments of inertia and L, M, N the components of aerodynamic moments with respect to the x_B, y_B, z_B body axes and I_{xz} represent a product of inertia, which is a measure of imbalance in the gliders mass distribution. The second and third term in each equation are the contributions due to gyroscopic motion, taking the gliders angular rates and mass distribution imbalance into account when calculating the net moments. The rolling, pitching and yawing moments L, M, N are calculated as follows

$$L = C_l Q S b \tag{22}$$

$$M = C_m Q S c \tag{23}$$

$$N = C_n Q S b \quad , \tag{24}$$

In the same way as was described for the coefficients C_Y and C_L, dimensionless quantities are added to the moments in equations (22) - (24) due to the gliders angular body rates.

More specifically, contributions are added to the coefficients C_l, C_m and C_n using the stability derivatives C_{l_p}, C_{l_r}, C_{m_q}, C_{n_p}, C_{n_r} and the dimensionless roll, pitch and yaw rates $\hat{p}, \hat{q}, \hat{r}$. The dimensionless moment equations then become

$$C_l = C_l(\beta, \delta_r) + C_{l_p}\hat{p} + C_{l_r}\hat{r} \tag{25}$$

$$C_m = C_m(\alpha, \delta_e) + C_{m_q}\hat{q} \tag{26}$$

$$C_n = C_n(\beta, \delta_r) + C_{n_p}\hat{p} + C_{n_r}\hat{r} \quad , \tag{27}$$

where the first terms are contributions due to side slip angle β, angle of attack α, as well as rudder and elevator deflections δ_r and δ_e respectively. The rest of the terms, like in the case of the forces, take into account the contributions due to the gliders own angular rates.

2) Optimal glide angle: The flight path angle γ of an aircraft is defined as the angle between the horizontal plane and the velocity vector, such that it describes whether the aircraft is climbing or descending in altitude (pitching up or down). For an aircraft gliding in steady linear flight without propulsion a horizontal distance d while descending a distance h in altitude, the glide angle γ' can be calculated as

$$\gamma' = \arctan(\frac{h}{d}) = \arctan(\frac{1}{d/h}) \quad , \tag{28}$$

where the quantity d/h is called the glide ratio which is desirable to maximize for any strictly gliding aircraft (as in no propulsion). Creating the hypotenuse out of the linear trajectory in a right-angled triangle with base d and height h, the lift Z will be normal to the trajectory and the drag X will parallel to the trajectory. Assuming a calm atmosphere and steady flight, the horizontal force equation becomes

$$Z \sin\gamma' = X \cos\gamma' \Rightarrow$$

$$\frac{\sin\gamma'}{\cos\gamma'} = \tan\gamma' = \frac{X}{Z} = \frac{C_D QS}{C_L QS} = \frac{C_D}{C_L} \quad , \tag{29}$$

where in the last step equations (6) and (8) have been used. Using equations (28) and (29), the glide angle can then be obtained as

$$\gamma' = \arctan(\frac{1}{d/h}) = \arctan(\frac{C_D}{C_L}) = \arctan(\frac{1}{C_L/C_D}) \quad ,$$

and since the glide ratio $d/h = C_L/C_D$ is the quantity to maximize, the optimal flight path angle is calculated as

$$\gamma_{opt} = -\arctan(\frac{1}{\max(C_L/C_D)}) \quad . \tag{30}$$

In equation (30) above a minus sign has been introduced because the flight path angle is defined as positive when the aircraft is climbing in altitude (i.e. nose is pitching up). Because the velocity vector is in the same direction as the x_W axis, and the pitch angle is defined as the angle between the x_B axis and the horizontal plane, the relationship between the optimal flight path angle γ_{opt} and optimal pitch angle θ_{opt} becomes

$$\theta_{opt} = \gamma_{opt} + \alpha_{opt} \quad , \tag{31}$$

in accordance with the relationships between reference frames as was discussed in section III-B. In equation (31) α_{opt} is the angle of attack where C_L/C_D assumes its maximum value.

D. Aerodynamic stability

The aerodynamic modes, closely connected to dynamic stability of an aircraft, describe how the aircraft behaves after being subjected to a perturbation following steady flight. That is, when submitted to a disturbance the aircraft tends to respond in accordance with the mode responsible for such changes. These modes are described by a modal shape, frequency and dampening where the latter two can be obtained from the poles of the system. The different modes are categorized in longitudinal and lateral modes [7].

1) Longitudinal modes: The longitudinal modes consist of two vertical oscillating motions, a long- and short-period oscillation referred to as the phugoid mode and the short-period mode respectively. The phugoid mode is an exchange of kinetic energy and potential energy and is characterized by being slow and lightly damped. The mechanism of the phugoid is that if the aircraft dives the speed increases and thus the pitching moment M increases since it is proportional to V^2. This in turn means the plane rises and slows down, decreasing the velocity and pitching moment before the next dive, completing one cycle. The short period is the faster of the two vertical oscillations, typically lasting only a couple of seconds, and is usually heavily damped. The motion depends primarily on rapid changes in pitch caused by a combination of variations in α, and the steep relationship between the pitching moment coefficient C_m and α.

2) Lateral modes: The lateral modes roll damping, dutch roll and spiral mode involve rolling and yawing motions and since these motions are interconnected, one mode often cause the other modes to appear. The roll damping mode is simply a damping of rolling motion. That is, when the aircraft has a rotation around the x_B axis the wing coming down will see an increase in angle of attack thus increasing the coefficient of lift C_L. The same applies to the opposite wing but with a decrease in angle of attack thus reducing the lift and creating a restoring moment opposite to the rotation. This effect can be amplified even more with the use of dihedral, that is installing the wings with an upward angle so that the wings are not aligned with the $x_B y_B$-plane. The dutch roll mode is a combination of roll and yaw motion. Consider a aircraft with a side slip angle β. Due to differences of drag and lift on each side of the aircraft creating a yawing moment the aircraft will turn thus decreasing β. The yawing moment will at the same time generate a roll angle, thus while decreasing the side slip angle the roll angle will increase causing the aircraft to overshoot the turn. The aircraft now once again have a side slip angle opposite to the original one and the cycle continues.

Spiral mode is a slow, often unstable, non-oscillatory mode. The cause of the spiral mode is due to some disturbance in roll or yaw from level flight which results in a non-zero side slip angle β. The side slip then causes wind to hit the tail, inducing a yawing moment which tends to increase the side slip, making matters worse. If left uncontrolled, the aircraft would then tend to slowly spiral toward the ground.

3) Loss of lift: An aircrafts coefficient of lift C_L depends on the angle of attack α. If one plots a graph of the relationship of C_L vs α a local maximum is obtained. That is, the coefficient of lift C_L increases as α increases until a certain α, in practice usually around $15°$, where it starts drastically decreasing. This region, where α has surpassed the point where the maximum is located, is called the stall region. In fact, stall is an aerodynamic condition defined as a loss of lift because of turbulent airflow on the top of the wing, resulting in unpredictable stochastic behaviour. That is, the stall region depends on the angle between the incoming wind and the angle of the chord line. In addition to this, the wings are usually mounted with an angle relative to the body called the angle of incidence, that is, the angle between the chord line of the wing and the x_By_B-plane. An incidence angle is used to minimize the angle between the body of the glider and the incoming wind in cruising flight. A consequence though is that the stall region increases and occurs for lower values of α thus restricting the manoeuvrability of the plane and increasing the risk of instability. To confirm that the aircraft is not in the stall region, one need to satisfy the following condition

$$15 > \alpha + \text{Incidence angle} \quad . \tag{32}$$

E. Control theory

1) Flight control: In flight control systems it is common to have stability augmentation systems which form inner loops of attitude control systems, where the attitude control systems form inner loops for path control systems. The term 'loop' is used here to denote a negative feedback loop with a proportional (P), proportional-integral (PI) or proportional-integral-derivative (PID) controller. Attitude control implies control of the angles pitch θ, roll φ and yaw ψ [8].

The inner-most loops for stability typically use P control on the angular roll, pitch and yaw rates p, q, r to dampen the effects of some of the aerodynamic modes of the aircraft, effectively causing commanded changes in attitude to be smoother. Having P control on the inner loop for the angular rate and P/PI control on the outer loop for the corresponding angle will have a similar effect as introducing a derivative part in the outer loop, i.e reducing oscillatory behaviour.

In pitch attitude control it is common to only use an elevator. Numerous constellations of different controllers could achieve pitch control, however a conventional controller can consist of having an inner P control loop with feedback of the pitch rate, and an outer P/PI loop with feedback of the pitch angle. With this setup, the complete controller would take the reference pitch as an input and output the appropriate elevator deflection command. Additionally, this setup increases the dampening of the short period mode due to the feedback of the pitch rate.

Lateral control of an aircraft is more complex than the pitch attitude (longitudinal) control, and in most modern aircraft the control is achieved through the use of both a rudder and ailerons. Specifically the roll angle is generally controlled via aileron deflections, however it can be induced (although more ineffectively) via a rudder deflection as well if the aircraft geometry allows it. A good technique for achieving spiral stability is to achieve dynamic stability in the roll angle, which requires a roll angle control system. The purpose of this system is to maintain roll attitude when disturbances occur and to respond swiftly and accurately to roll commands from the guidance system (or pilot if the aircraft has one). Such a system can typically be realized by a control loop using negative feedback on the roll angle, and if additional stability is required, an inner loop for roll rate may also be worth considering.

The direction control systems purpose is to allow automatic steering towards some desirable direction. This can be achieved with a control system that uses feedback of the heading and bank angle, that is, the angle between north and the direction of travel and the angle between the wings and horizon (if the wings protrude in the x_By_B plane with zero dihedral). One way to structure such a control system is by having an outer loop with feedback on the heading angle, and the heading controller output is fed to an inner loop, serving as a reference in bank angle. With appropriate gains for the inner and outer loop controllers it is then possible to track a desired heading whilst maintaining aerodynamic stability since the inner loop makes sure the bank angle is kept within certain limits, because the feedback of the bank angle effectively reduces the associated control surface deflection(s) even if the heading error is large.

When control surfaces are commanded to turn it causes a hinge moment on the surface due to the oncoming wind, where the size of the moment depends on the angle of deflection. Because the actuators take some finite time to turn the control surfaces, and because the moments do not increase linearly with the angle of deflection, it is then important to have reasonable models of the actuators if the system dynamics are to be accurately represented in a closed loop system with attitude control. It is then considered a good idea that the mathematical actuator models contain non-linear characteristics and be represented by at least second order transfer functions, as described in [8].

2) Sensor fusion and filtering: A Kalman filter is an optimal estimation algorithm commonly used in navigation and guidance systems. It is often used when the states of interest are not directly measurable and/or measurements can be obtained from several different sensors that may have been subject to noise. The process of combining sensor data is called sensor fusion. The Kalman filter assumes that the process to estimate

is of the form

$$\dot{x} = Ax + Bu + w \quad , \tag{33}$$

where x and u denote the state value and system input/control effort respectively, w is assumed to be white noise with some known covariance [9]. It is further assumed that measurements taken of the process occur according to

$$z = Cx + v \quad , \tag{34}$$

where z is a sample, C is the ideal connection between the state value and the measured value and v is the measurement error. Assuming no control inputs u and using $\hat{x}$ to denote the estimate of the state x, the process (33) - (34) can then be discretely modeled as:

$$\hat{x}_k = A_k\hat{x}_{k-1} + w_k \tag{35}$$

$$z_k = C_k\hat{x}_k + v_k \quad . \tag{36}$$

Using $\hat{x}_k^-$ to denote the predicted state estimate prior to receiving the measurement at time t_k, the system error is defined as

$$e_k^- = x_k - \hat{x}_k^- \quad , \tag{37}$$

and the predicted (a priori) estimate itself is calculated using the last iterations updated estimate $\hat{x}_{k-1}^+$ as

$$\hat{x}_k^- = A_k\hat{x}_{k-1}^+ \quad . \tag{38}$$

To obtain the updated estimate $\hat{x}_k^+$ (a posteriori), a linear mix of the measured value and the prior best estimate $\hat{x}_k^-$ is used

$$\hat{x}_k^+ = \hat{x}_k^- + K_k(z_k - C_k\hat{x}_k^-) \quad , \tag{39}$$

where K_k is the so called Kalman gain which determines how much weight is given to the error between the best estimate and the actual measured value. The Kalman gain gets updated over time, computed by the equation

$$K_k = \frac{P_k^- C_k^T}{(C_k P_k^- C_k^T + R_{v,k})} \quad . \tag{40}$$

The error covariance matrix P_k is defined at time t_k as

$$P_k = E[e_k e_k^T] = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T] \quad . \tag{41}$$

Manipulating equation (41) using some linear algebra eventually yields equations (42) - (43) below for the predicted and updated error covariance matrices respectively

$$P_k^- = A_k P_{k-1}^+ A_k^T + R_{w,k} \tag{42}$$

$$P_k^+ = (I - K_k C_k)P_k^- \quad . \tag{43}$$

In equations (40) - (43) above E is the expected value, I is the identity matrix and $R_{v,k}$,$R_{w,k}$ is the covariance matrices of v and w respectively [10].

F. Spherical geometry

In geometry the geodesic is defined as the shortest route along a spheroid surface between two points and an azimuth is an angle used when expressing a direction relative to some reference direction. Using true north as the reference direction, and assuming the earth has a spherical shape, the geodetic azimuth Γ can be calculated as:

$$\Gamma = \arctan\left(\frac{\sin(\Delta\lambda)\cos(\varphi_2)}{\cos(\varphi_1)\sin(\varphi_2) - \sin(\varphi_1)\cos(\varphi_2)\cos(\Delta\lambda)}\right) \quad . \tag{44}$$

The geodetic azimuth is the angle between true north and the geodesic, that is, the compass heading one should initially follow from point one (φ_1, λ_1) in order to travel along the shortest path to point two (φ_2, λ_2). In equation (44) the positions are given in latitude and longitude φ and λ respectively and $\Delta\lambda = \lambda_2 - \lambda_1$ denotes the difference in longitude between the two points [11].

IV. Methodology

This section aims to demonstrate and motivate the solutions used or proposed in this project for the development and implementation of the flight control system.

Designing a control system is generally an iterative process, where the closed-loop system response often needs to be analyzed and the controllers re-tuned accordingly, as such it is preferable if the initial control system to be tested works relatively well. When the system to be controlled does not exist physically or the ability to do real life testing is limited, as is the case in this project, designing a realistic simulation environment is of utmost importance if the control system is to exhibit desired characteristics when it comes to real life testing. In light of this, a simulation environment was developed in SIMULINK to aid in the development of the flight control system.

Even with a capable physics engine and tools for control analysis in a simulation environment, it serves as a poor platform for controller tuning without a realistic model of the glider. The model of the glider is obtained from XFLR5 data. XFLR5 is a software which BSc project I_1 uses to design the gliders geometry such as wings, tail, control surfaces, placement of centre of gravity and so forth. The software is also capable of performing different kinds of analyses, where all force and moments coefficients in table II are obtained by analyzing how the forces and moments change when the glider is subject to different angles of attack α, side slip angle β and control surface deflections δ.

In section II-A the gliders physical design was presented and the control surfaces explained. Even though the design has control surfaces mounted at an angle on an inverted V-tail (sometimes referred to as ruddervators), in this project both ruddervators (RVs) are treated as a virtual elevator and rudder. The reason behind this is discussed in section VI-B. They can be treated as a virtual elevator and rudder so long as the ruddervators are coupled as such when performing the

XFLR5 force and moments analyses, meaning that as seen from behind both ruddervators are deflected up or down at the same time or right and left at the same time. This way the elevator and rudder effects of the control surfaces can be obtained and the XFLR5 data treated as if the glider had a conventional inverted T tail.

A. The SIMULINK environment

The simulation environment was created in SIMULINK which is an extension to MATLAB. One of the great advantages of using SIMULINK is all the predefined blocks and tools available to aid in control system analysis and simulating flight dynamics. In addition, it offers a tool for translating and exporting the desired SIMULINK model in to C code to be compatible with the on-board microcontroller. So in theory, the control system can be designed and tuned according to simulated flight behaviour and then translated into C code for the real life implementation.

1) Input: In addition to the design parameters provided in table I the simulation environment requires information about all the aerodynamic coefficients of the glider, the stability derivatives and its moments of inertia, see table II. The complete data set presented in table II are obtained through XFLR5 via BSc project I_1. The different angles and deflections used for analyses in XFLR5 are also of interest in the simulation environment in order to handle the given data correctly, thus this also needs to be passed as an input, see table III.

TABLE II
INPUT REQUIREMENTS

Drag coefficient C_D	Side force coefficient C_Y	Lift coefficient C_L
Rolling moment coefficient C_l	Pitching moment coefficient C_m	Yawing moment coefficient C_n
Stability derivative C_{Y_p}	Stability derivative C_{Y_r}	Stability derivative C_{L_q}
Stability derivative C_{l_p}	Stability derivative C_{l_r}	Stability derivative C_{m_q}
Stability derivative C_{n_p}	Stability derivative C_{n_r}	Inertia I_{xx}
Inertia I_{yy}	Inertia I_{zz}	Inertia Ixz

TABLE III
SWEEP VECTORS WITH VALUES USED FOR XFLR5 ANALYSIS

Rudder deflection vector δ_{rudder}	Elevator deflection vector $\delta_{elevator}$	Angle of attack vector α_{sweep}	Side slip vector β_{sweep}

With the inputs described in tables II and III the coefficients for forces and moments are then structured in either m x n or k x l matrices, where m and k are the number of elements in α_{sweep} and β_{sweep}, and n and l are the number of elements in δ_e and δ_r. This structure essentially allows for forces and moments to be obtained from matrices for different angle of attack α or side slip angle β by the row, and different control surface deflections by the column.

2) Physics engine: The physics engine in the simulation environment is responsible for calculating all the forces and moments acting on the glider. The aerodynamic force and moment contributions due to angle of attack α, side slip angle β and control surface deflections δ_e/δ_r are obtained by using a pre-lookup with linear interpolation on the matrices of data described in section IV-A1. The pre-lookup and linear interpolation is performed by predefined SIMULINK blocks, and the structure is such that α, β, δ_{rudder}, $\delta_{elevator}$ is fed back through the simulation such that the appropriate force and moment is obtained from it depending on the gliders current state. At this stage a check is made to see if the glider is in the stall region and the lift coefficient C_L needs to be zeroed or not (see section III-D3). The SIMULINK model does this via a MATLAB function that checks whether the condition in equation (32) is satisfied based on the current α and thus passes C_L through and sets C_L to zero if its not. More specifically, this is the way the first terms in the dimensionless aerodynamic equations (17) - (18) and (25) - (27) are obtained, and the rest of the terms (stability derivative contributions) are added on top of this using trivial arithmetic and the current angular rates p, q, r as well as the current velocity V. It is a straightforward matter then to realize equations (6) - (8) and (22) - (24) at this point, where the dynamic pressure Q is calculated according to equation (9). The air density ρ is obtained via a predefined SIMULINK block which only takes the current altitude as an input and implements a mathematical representation of the international standard atmosphere (ISA) model [12].

The aerodynamic forces calculated above are in the XFLR5 reference frame F_C, though it is common practice in aerodynamics to have them defined in the wind axis reference frame. In addition, the simulation environment has a predefined SIMULINK 6 Degrees of Freedom Wind Angles (6DOF) block which requires its force inputs to be in the wind axis reference frame F_W. Thus the forces in the x_C and z_C axes needs to be multiplied with -1 and the y_C axis multiplied with a body to wind transformation matrix T_{BW}, effectively rotating all force contributions into the F_W reference frame. The rotation matrix T_{BW} itself is calculated with the help of the current α, β (outputs from the 6DOF block) and a predefined SIMULINK block. In addition to the aerodynamic forces, the force of gravity also needs to be added. To add the force of gravity W in the wind axis reference frame it is first defined as a vector $W_V = [0, 0, mg]^T$ in the vehicle-carried NED frame F_V where gravity is aligned with the z_V axis and then rotated via the transformation matrix T_{EW}, where the matrix T_{EW} is obtained as an output straight from the 6DOF block. To be clear here, the force of gravity should get rotated by a F_V to F_W rotation matrix T_{VW} but the end result is the same.

$$W_W = T_{VW}W_V = T_{BW}T_{VB}W_V = \quad (45)$$

$$= T_{BW} \begin{bmatrix} -mg\sin\theta \\ mg\cos\theta\sin\varphi \\ mg\cos\theta\cos\varphi \end{bmatrix} = \quad (46)$$

$$= T_{BW}T_{EB}W_V = T_{EW}W_V = W_W \quad (47)$$

The reason for this is the fact that the third column of the matrices T_{EB} and T_{VB} are identical as discussed in section III-B. As can be more clearly seen in equation (46), this effectively implements the force of gravity according to the force equations (6) - (8). The moments obtained from the XFLR5 data however are already defined in the body-fixed reference frame F_B.

The forces and moments are then passed as inputs to the 6DOF block which implements the equations (3) - (5) and (19) - (21). Although not all force and moment contributions are supplied to the block on purpose. The reason for this is because the last terms in equations (3) - (5) and the last two terms in equations (19) - (21) related to gyroscopic motion are calculated by the block internally. To calculate these contributions the mass m and the inertia tensor I needs to be defined in the block parameters. Specifically the inertia tensor is passed as

$$I = \begin{bmatrix} I_{xx} & 0 & -I_{xz} \\ 0 & I_{yy} & 0 \\ -I_{xz} & 0 & I_{zz} \end{bmatrix} \quad ,$$

which implements the final two terms in equations (19) - (21) via the 6DOF blocks internal computation. In addition to all parameters defined in the block, the 6DOF block have multiple outputs, some of which are fed back to be used in the calculations of forces and moments. These are angle of attack and side slip angle for the pre-lookups, body rates for the stability derivative contributions as well as the current speed and altitude. The speed and altitude are not direct outputs from the 6DOF block. The speed is simply calculated as the magnitude of a output velocity vector and the altitude is obtained by multiplying the output z_E coordinate by -1 (since z_E is positive down). Note here that only the 6DOF block outputs of relevance to this thesis are mentioned here. Furthermore, the initial state of the glider is defined in 6DOF block. That is, its starting position in the earth fixed NED F_E reference frame, initial velocity V_0, initial angle of attack α_0, initial side slip angle β_0, initial orientation and initial body rotation rates.

To be able to study the behavior of the glider and to control it, its orientation with respect to the earth must be derived. The orientation is defined by the inertial NED to body rotation matrix T_{EB}, where a predefined SIMULINK block can be used to extract the angles of rotation from the matrix itself (assuming the aerodynamic rotation sequence). These are the yaw Ψ_E, pitch θ and roll φ angles and it should be noted that the yaw angle Ψ_E is the angle between true north and the x_B axis. The matrix T_{EB} itself is calculated as

$$T_{EB} = T_{WB}T_{EW} = T_{BW}^T T_{EW}$$

3) SIMULINK implementations of on-board hardware: To bring the simulation environment even closer to reality and make it a more suitable platform in which to tune the flight control system, some of the hardware inherited for the project is implemented in SIMULINK. See section II-B for a discussion about the on-board hardware.

The digital implementation for the GPS module is really an implementation of the UBX-NAV-PVT message which contains the outputs deemed necessary for this project in combination with the sensors data [13]. Specifically the current longitude, latitude and heading due north is of interest in this project. The GPS message has several outputs that are believed to be of future interest but are ignored within the scope of this project, such as the ground speed, height above mean sea level and NED velocities.

The GPS is realized with the help of a predefined MATLAB object gpsSensor() and a couple of predefined SIMULINK blocks which can convert positions from the inertial F_E reference frame to longitude, latitude, altitude coordinates in the LLA reference frame, and also perform the inverse of this operation. The gpsSensor() object outputs noisy LLA coordinates, NED velocities, ground speed and the current heading. In order for the gpsSensor() object to work it needs the current position and velocity expressed in the F_E reference frame, which it receives from the 6DOF block at a frequency of 2 Hz (which is the max. output rate for the UBX-NAV-PVT message from the physical GPS module).

In addition, it should be mentioned here that the gpsSensor() object considers a NED reference frame where it mathematically flattens the surface of the earth around its choice of origin and because of this, positions far from its origin generate incorrect LLA coordinates. Thus, the origin of the gpsSensor() object is set to Esrange. For the current inertial position provided to be compatible with the gpsSensor() it then needs to be converted into a NED F_E coordinate system with its origin also at Esrange, which can be done with the use of the SIMULINK block 'Flat Earth to LLA' and its inverse (redefining the place of origin in the process). The reason this is needed is due to the fact that when the simulation starts, the default origin of the F_E reference frame is set to $0°$ longitude and $10°$ latitude and this reference frame is also used to set the initial position of the glider in the 6DOF block.

Aside from the digital implementation of the GPS message, there is also one such implementation for the on-board sensors. For this an Inertial Measurement Unit (IMU) block is used, once again predefined in SIMULINK. The IMU block takes the angular rates and linear acceleration in the F_E reference frame as well as the rotation matrix T_{EB} as inputs at 100 Hz and outputs noisy data at the same frequency as if it was actually measured by the on-board gyroscope and E-compass (accelerometer and magnetometer). Furthermore,

the IMU block was tuned by adjusting its parameters in the form of biases, sensitivities and noise according to the data sheets of the on-board sensors [14], [15].

The frequency 100 Hz was chosen in order to have a fast update rate for controller performance while still being slow enough to not reach the maximum baud rate (115200 bit/s) of the UART line used to send the sensor data from the on-board FPGA to the microcontroller. In addition, all the sensors described in section II-B have the option of 100 Hz update rate. The idea was to send sensor data of both gyroscope and E-compass in all axes as one data package, totaling 144 bits. Sending the sensor data through to the microcontroller then requires 14400 bit/s which allows for plenty of baud rate left for additional communication.

In addition to the SIMULINK implementations of the GPS and sensors, the model also contains nonlinear second order actuators in the form of predefined SIMULINK blocks in accordance with what was discussed in section III-E1. More specifically the block parameters allow for setting how fast the control surfaces can move and what angles the control surfaces can deflect to, which was set to $\pm 20^\circ$ max. deflection and 60°/s max. speed according to an estimation from BSc project I_4.

4) Sensor fusion and filtering implementation: For sensor fusion and filtering the predefined Attitude and Heading Reference System (AHRS) SIMULINK block is used along with a block that translates rotation matrices into the corresponding rotation angles. More specifically, the predefined AHRS blocks orientation output is the rotation matrix T_{VB} as described in section III-B. The AHRS model used in this project then takes simulated sensor data from the IMU block described in section IV-A3 as inputs (or actual sensor readings) and outputs sensor fused and filtered data in the form of orientation angles Ψ, θ, φ and angular rates p, q and r. The motivation behind this is the fact that the AHRS block has an optimal estimation algorithm implemented in the form of a discrete indirect Kalman filter and is compatible with the SIMULINK to C code export tool. The Kalman filter implementation models a process according to equation (35) where x_k is a 12x1 error process vector

$$\hat{x}_k = [O_k, b_k, a_k, d_k]^T = A_k \hat{x}_{k-1} + w_k \quad , \tag{48}$$

and O_k, b_k, a_k, d_k correspond to the (3x1) orientation error, gyroscope zero angular rate bias, acceleration error and magnetic disturbance error respectively. The matrix A_k models the state transition and w_k models additive noise. Because $\hat{x}_k$ is an error process, the predicted state estimate $\hat{x}_k^-$ is zero and by extension so is the matrix A_k in accordance with equation (38). More specifically, the Kalman equations implemented by the AHRS block algorithm are reduced versions of the Kalman equations (38), (39), (42) described in section III-E2, computed as

$$\hat{x}_k^- = 0 \tag{49}$$

$$\hat{x}_k^+ = K_k z_k \tag{50}$$

$$P_k^- = R_{w,k} \quad , \tag{51}$$

and the Kalman equations (40), (43) are computed according to their earlier definitions. For a more detailed overview of the AHRS block Kalman filter algorithm, the reader is referred to [16].

5) Heading reference: To steer the glider towards Esrange, the geodetic azimuth Γ is calculated according to equation (44). The geodetic azimuth, the angle between true north and the nose of the glider (x_B axis), is the initial heading the glider should have in order to travel the shortest distance possible toward Esrange. Using equation (44) for the heading logic is motivated then by minimizing the total fly time of the glider, which is desirable from several standpoints. Since the glider has no propulsion and ability to regain altitude, it must travel toward its destination as swiftly as possible in order to make it there before it runs out of altitude. The current longitude and latitude coordinates are passed as inputs from the plant and the longitude and latitude coordinates for Esrange are defined. Once the geodetic azimuth angle is calculated it is passed along with the current heading relative true north Ψ_E and the current heading relative magnetic north Ψ measured by the digital magnetometer to a MATLAB function that calculates the reference value for yaw. The value is obtained by calculating a turn angle $\Delta\Psi$ that is added to the current heading,

$$\Delta\Psi = \Gamma - \Psi_E$$

$$\Psi_{ref} = \Psi + \Delta\Psi \quad .$$

In addition, the turn angle $\Delta\Psi$ is limited to $\pm 30^\circ$ which was implemented after observing spiral instability due to large rudder deflections when $\Delta\Psi$ took on its largest possible unsaturated values of around $\pm 180^\circ$. Note that the control could have been operated by the GPS alone since it provides a current heading relative true north and the desired heading can be obtained because the geodetic azimuth Γ can be calculated. Though, because the frequency of the GPS is only 2 Hz this would limit the control system making it unnecessarily slow. If the magnetometer reading is used to track the heading of the glider instead which has a frequency of 100 Hz the speed of the controller is increased significantly. Even though the magnetometer is used for tracking the heading, the GPS offers the only way to measure the heading relative true north (like equation (44) calculates a heading due true north) and is thus used to set the desired heading. The end result of this is that the frequency of the tracking operates at 100 Hz to allow for faster controllers meanwhile the heading reference Ψ_{ref} is updated at a frequency of 2 Hz.

6) Pitch reference: In light of what was discussed in section III-C2, the pitch reference is calculated using equations (30) and (31)

$$\theta_{ref} = \theta_{opt} = -\arctan\left(\frac{1}{\max(C_L/C_D)}\right) + \alpha_{opt} \quad .$$

The motivation for using this formula comes from the desire to achieve maximum glide ratio, minimizing the possibility of crashing into the ground before reaching the destination. It is desired to keep as much of the altitude as possible until initiating landing sequence once the glider is in the vicinity of Esrange. That is because once the glider drops in altitude, due to disturbances or air pockets etc, there is no way in regaining it again which might cause the glider to not reach all the way to Esrange. Using the XFLR5 data for neutral control surface deflections, it can easily be found that the maximum value of $C_L/C_D = 8.35$ is obtained for $\alpha_{opt} = 3^\circ$ which yields the desired pitch reference

$$\theta_{ref} = -3.83^\circ \quad .$$

7) Control system: The control system developed in SIMULINK follows a similar structure to what was described in section III-E1. Negative feedback of the pitch rate q and roll angle φ is used for smoother control and stability, forming inner loops for the pitch attitude and heading control loops, see Fig. 5. Note here that the control system outputs the commands for the virtual elevator and rudder. The block that acts on the yaw error signal $\Psi_{ref} - \Psi$ is there to handle the case where the yaw angle goes from 180° to -179°. Because of the 2 Hz update time from the SIMULINK GPS module which the heading reference system is dependent on (see section IV-A5), the yawing error can become overly large for half a second during the transition which causes a momentary but undesirably large rudder deflection. All controllers are discrete and operate on the same frequency as the assumed sensor update rate of 100 Hz using no anti-windup. The discrete SIMULINK control blocks are realized using the forward and backward Euler methods for integral and derivative approximations in the z-domain with a transfer function of the form

$$P + IT_s\frac{1}{z-1} + D\frac{1}{T_s}\frac{z-1}{z} \quad . \tag{52}$$

Here P, I and D are the gains for the controllers proportional, integral and derivative parts respectively and $T_s = 0.01$ s is the sample time. The controllers were tuned in part by using the PID tuner tool available in SIMULINK which creates a linearized version of the entire model and provides a convenient user interface where step responses, Bode plots amongst other things can be analyzed for different controller parameters. While the built-in PID tuner tool provides adequate results most of the time, the controllers were also re-tuned according to time domain responses during simulated test flights which is necessary since the linearized model created by the SIMULINK tool is not entirely satisfactory. The gains used in the control system are seen in table IV. Note that all derivative gains are zero, the reason being that the performance of the setup proposed in Fig. 5 was deemed better than with PID control, see sections V-B, VI for results and discussion.

V. SIMULATION RESULTS

The results presented in this section have been produced without the use of the IMU and AHRS SIMULINK models introduced in section IV-A and instead the orientation angles Ψ_E, θ and φ are obtained straight from the T_{EB} rotation matrix (note that Ψ_E implies the heading due true north since no magnetometer reading is available). See section VIII-C for further discussion about the IMU/AHRS models.

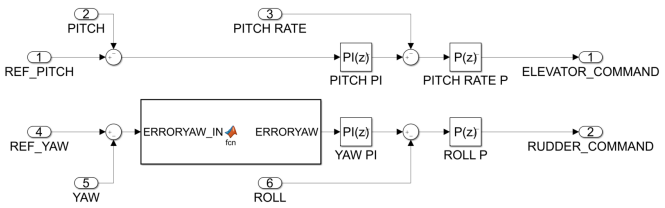


Fig. 5. The control implementation in the simulation environment.

TABLE IV
CONTROLLER GAIN PARAMETERS

Controller	P	I	D
PITCH RATE P	-0.9654	0	0
PITCH PI	0.8703	0.2196	0
ROLL P	-0.4245	0	0
YAW PI	0.4068	0.001020	0

A. Open loop

The open loop of the simulation environment is the system described in section IV-A without any control or feedback, meaning it is the physics engine running the simulation using only neutral control surface deflection data from some initial condition. To test the open loop system it is compared to the results obtained by the software XFLR5 used for the geometry design of the glider. Specifically, the aerodynamic modes and steady state flight conditions achieved in the open loop SIMULINK model is compared to those predicted by XFLR5 by comparing pole-zero maps and the steady state flight conditions.

To confirm the validity of the steady state flight conditions the open loop is initiated from several different starting conditions with the initial speed, pitch and angle of attack being the varying variables, see table V. The results of the open loop analysis can be seen in Fig. 6 - 9.

TABLE V
INITIAL CONDITIONS FOR OPEN LOOP SIMULATION

	Velocity [m/s]	Pitch [degrees]	Angle of attack [degrees]
Run 1	20	0	0
Run 2	35	0	0
Run 3	40	-30	0

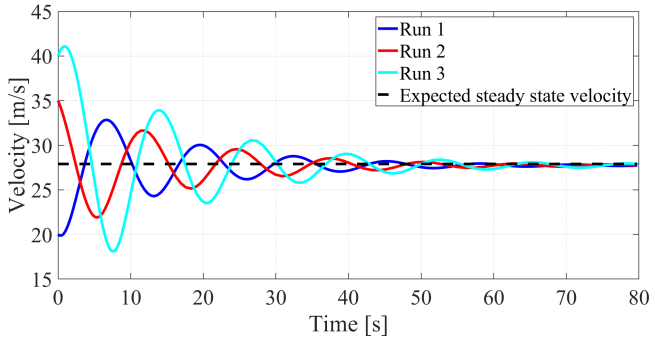


Fig. 6. Velocity converging to the same steady state value for three different initial conditions.

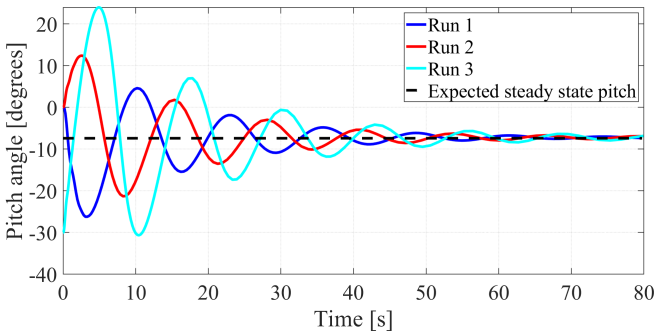


Fig. 7. Pitch converging to the same steady state value for three different initial conditions.

The comparison of the pole-zero maps can be seen in Fig. 9 and the modes frequencies and dampening are given in table VI. The poles in the figure describe the different modes of the glider, see section III-D. In addition to the poles predicted by XFLR5, there are two more poles in the simulation environment close to the origin. These are mathematical poles caused by integrators in the 6DOF SIMULINK block and are not part of the gliders flight dynamics.

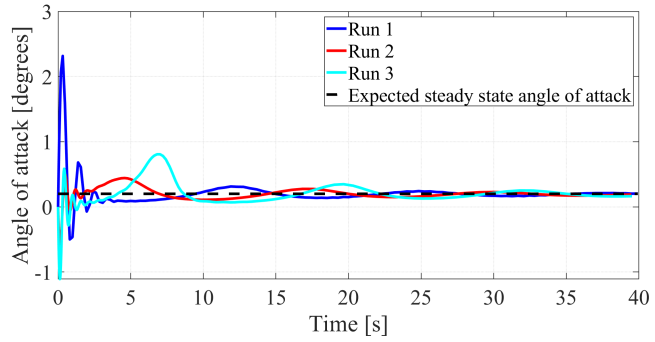


Fig. 8. Angle of attack converging to the same steady state value for three different initial conditions.

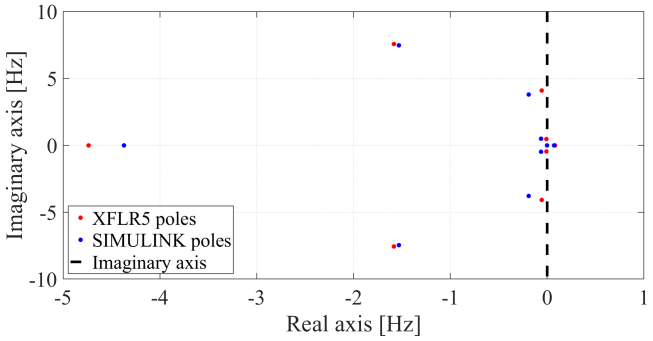


Fig. 9. Pole-zero map comparison between XFLR5 and the simulation environment.

TABLE VI
POLE FREQUENCY AND DAMPENING

SIMULINK poles, XFLR5 poles	Frequency [rad/sec]	Relative dampening
Spiral mode	0.070, 0.083	1, 1
Phugoid modes	0.492, 0.475	0.128, 0.019
Dutch roll	3.800, 4.080	0.050, 0.014
Short period	7.610, 7.400	0.201, 0.205
Roll dampening	4.370, 4.735	1, 1

B. Controllers

The glider control system can be implemented in many different ways. In Fig. 10 and 11 comparisons between two types of control laws are shown for pitch and yaw control respectively. In both simulations the initial conditions are set to steady flight conditions $V_0 = 27.9$ m/s, $\alpha_0 = 0.2°$ and $\theta_0 = -7.48°$.

In Fig. 10, a PID pitch angle controller is compared to the performance of having PI control on pitch angle and an inner loop with P control on the pitch rate (where the latter control law is shown in Fig. 5). Both control laws track the reference pitch θ_{ref}.

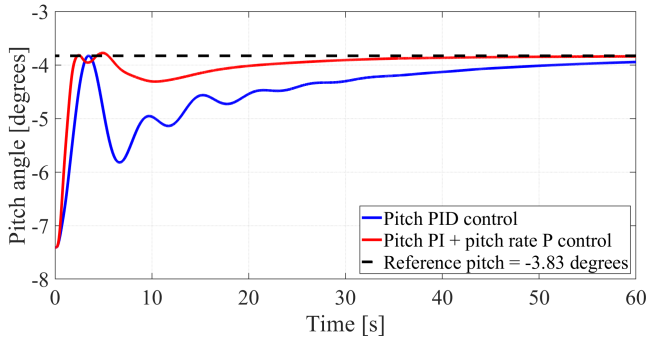


Fig. 10. Step response of two different implementations of pitch control.

Just like in Fig. 10, Fig. 11 shows a comparison between two control laws. One has a yaw angle PID controller and the other control law has PI control on the yaw angle with an inner loop for P control on the roll angle (where the latter control law is shown in Fig. 5). The reference yaw goes from

$\Psi_{ref} : 0° \rightarrow 30°$ at time $t = 10$ s. In the figures it can be seen that a PI controller on the outer loop and a P controller on the inner is faster and more robust than a PID controller. The PID controller even diverges from the reference value when controlling yaw as seen in Fig. 11, where the glider eventually starts to spiral toward the ground.

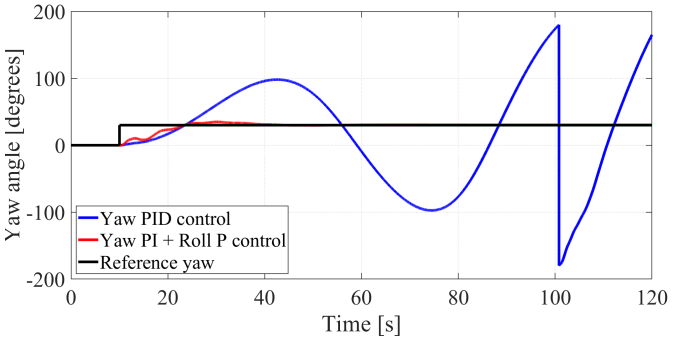


Fig. 11. Step response of two different implementations of yaw control.

C. Tracking to Esrange

In Fig. 12 and 13 one can see the glider traveling approximately 70 km horizontal distance to Esrange. The initial position is a GPS position obtained from a FFU from a previous launch and the initial altitude is set to 10 km. Additionally, the initial heading was set to a 'worst case scenario' such that the initial turn angle necessary would be around 180°. Note that the "tornado" at the end of the trajectory in the 3D plot is the glider passing Esrange and adjusting accordingly, spiraling down until it hits the ground. The controllers used during the flight test were the ones seen in Fig. 5 described in section IV-A7. In Fig. 14 one can see how long it is estimated by the simulation environment before the glider reaches Esrange and when it is estimated to hit the ground. In addition, to confirm that the stall region is not reached Fig. 15 is included.

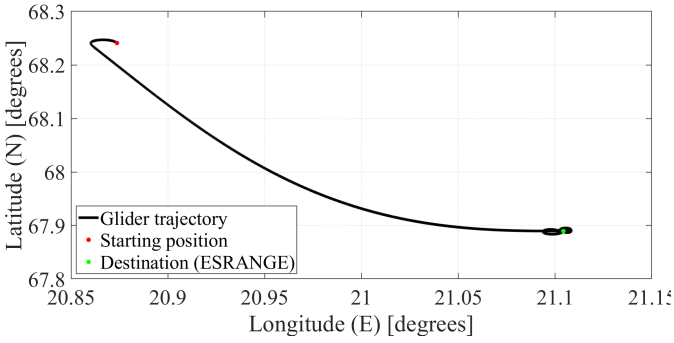


Fig. 12. Longitude and latitude plot of the glider trajectory to Esrange, initialized at 70 km horizontal distance away at 10 km altitude.

D. The impact of the altitude on the controllers

Last but not least, it was noticed that the controllers performed differently depending of the altitude of the glider, which in turn affects the speed of the glider because of the varying air density ρ. In order to study the mentioned

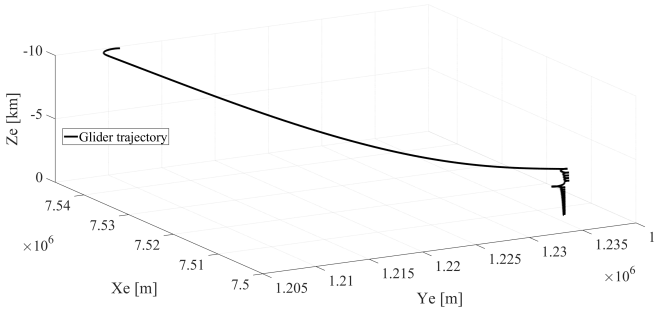


Fig. 13. 3D plot of the glider trajectory to Esrange, initialized at 70 km horizontal distance away at 10 km altitude

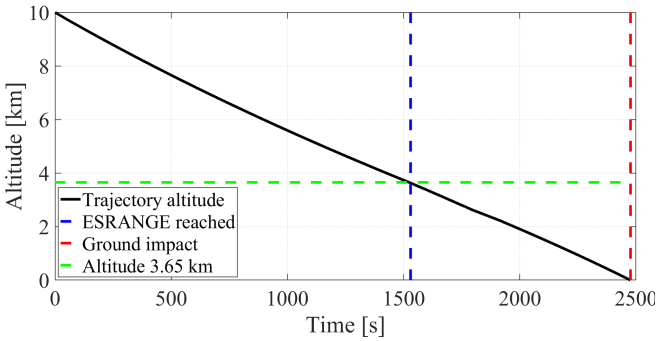


Fig. 14. Altitude of the glider while traveling toward Esrange.

behaviour the controllers were tested with the height initiated at the two extreme points of the flight altitude, that is 10 km and sea level, and the time domain response was studied. As for the simulated flight test, the controllers used for this test were the ones seen in Fig. 5. The results for the pitching is presented in Fig. 16 and for yawing in Fig. 17. The aim is to highlight the dependency and effects of height on the controllers. In Fig. 16 the variation of the pitch control is small compared to the difference in yaw control seen in Fig. 17 and as such this should be considered when finalizing the controller.

VI. Discussion

In this section the results and some of the choices, methods, and assumptions of this project will be discussed.

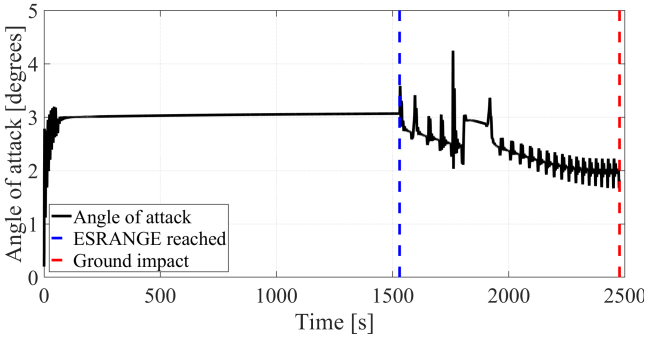


Fig. 15. The angle of attack of the glider while traveling toward Esrange.

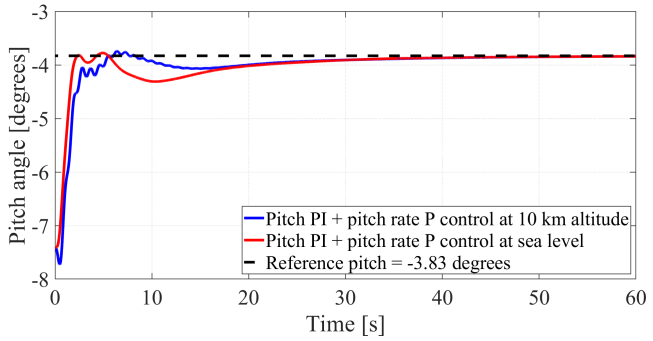


Fig. 16. Comparison in step response at sea level vs. 10 km altitude

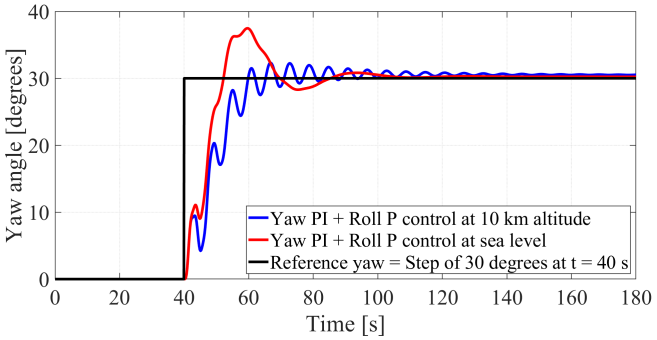


Fig. 17. Comparison of step responses at sea level vs. 10 km altitude

A. Discussion of simulation results

The steady state conditions and pole placements as seen in Fig. 6 - 9 and table VI is consistent with the predictions made by XFLR5. Specifically, Fig. 6 - 8 show that the pitching moment converges to zero around the same velocity, pitch and angle of attack as is predicted by XFLR5 and table VI shows what is deemed to be closely matching frequencies of the aerodynamic modes predicted by the two simulation environments (although note that the relative dampening does not match that well). Because the two simulation environments predict similar dynamics and steady states it is believed that the SIMULINK model has a physics engine which is (at least) similar to the one XFLR5 uses. So in order to judge if the SIMULINK environment is a good model of reality, it comes down to the credibility of the XFLR5 physics engine. XFLR5 is an analysis tool used in the aerospace community for airfoil, wing and plane design which have been maintained and updated several times since its initial release in 2003 [6]. As such, the results of XFLR5 are considered credible.

From Fig. 10 and 11 the project members argue that the control laws utilizing PI control for the outer loops and P control for the inner loops are more suitable for the task because it is faster and less oscillatory (for the yaw the PID controller is even unstable). These results are also in agreement with the theory presented in section III-E1.

Using the controllers from Fig. 5 the heading reference system could be tested while also controlling for maximum glide ratio. As is seen in Fig. 12 - 15, the glider successfully reaches Esrange along a reasonable path well before ground impact. The altitude is seen to be steadily declining during the trajectory (even when the glider starts circling Esrange) and the glider also successfully avoids the stall region as can been seen in Fig. 15.

The results in Fig. 16 and 17 implies that a control law tuned for sea level conditions have different transient behaviour at different altitudes. The settling times are similar but there are some differences in oscillations and overshoot. The oscillations is expected to have a major impact on the amount of energy required to turn the actuators, because it causes unnecessary and undesirable turning back and forth of the actuators during the first minutes of flight. Because of the limited space available in the CU for storing batteries, the energy consumption during flight is already a pressing issue and as such the oscillations even less desirable.

B. Virtual elevator and rudder

To use XFLR5 data containing force and moment contributions due to the individual ruddervators and design the SIMULINK environment accordingly is preferable to using the virtual elevator and rudder design that was demonstrated in this thesis. Using a virtual elevator and rudder design works better if one is only interested in simulating and controlling either only pitch or only yaw. This is because then the RVs always act synchronized, pitching up or down or yawing left or right and the XFLR5 data used to obtain the forces and moments should serve as a sufficient model. When a pitching and yawing action are combined however, using the virtual elevator/rudder design becomes a worse model to use when hoping to control RVs in reality. The simulation assumes that the elevator and rudder move entirely independent from each other which can produce deflections that are impossible to achieve with RVs in reality. In addition, the combined effects of an elevator and rudder and the combined effects of two RVs do not produce the same force and moment contributions, making it far more preferable to handle the data for the RVs independently when wishing to control RVs in real life. The transition from a SIMULINK model with an RV architecture to an elevator/rudder architecture was easy since earlier iterations of the glider design proposed by BSc project I_1 used a conventional plane tail and as such a model with that design already existed. See section VIII regarding future work if a ruddervator design for the glider is to be used along with a virtual elevator/rudder SIMULINK design, although it is not recommended by the authors of this thesis.

The decision to use a simulation environment with an elevator/rudder architecture instead of a RV architecture was made due to bugs in the SIMULINK model that was developed to take force and moment contributions from each individual ruddervator independently. Many painstaking hours were spent troubleshooting, but the problem remains unsolved at the time of writing this thesis. The problem might be due to a mathematical/numerical error or an error in the adding of

force/moment contributions from the right and left ruddervator. The problem reveals itself either when an open loop simulation is run with the control surfaces set to some symmetric but none-zero value or when the control surface deflection commands are subject to a step or pulse in the otherwise open loop system. The glider then exhibits a strange behaviour where the yawing moment (which should be zero in open loop or under symmetric RV pitching deflections) grows exponentially if one allows the simulation to continue for several minutes. This is *not* believed to be caused by the spiral mode described in section III-D due to the fact that no initial side slip or roll angle is given to the glider and all yawing and rolling moments should be zero. Troubleshooting have not revealed the culprit of this problem, but it is suspected to be connected to the RVs in some way. The reason for this is the fact that the zero deflection (open loop) simulation exhibit expected results whereas it does not when deflections are thrown in the mix. Also, this problem has not been detected when running an elevator/rudder design which seems to indicate an error either in how the data from XFLR5 is handled or how the RV contributions are added in SIMULINK. It is possible it has to do with the numerical data from XFLR5 not being entirely symmetric (upon close inspection) where it should be due to the gliders geometric symmetry.

C. Assumptions made in the simulation

The simulation environment assumes a perfectly symmetrical glider in the $x_B z_B$-plane and a calm atmosphere. The glider symmetry is a design choice and is utilized to simplify the moment equations (19) - (21). In reality, the mass distribution will be dependent on many factors such as the final choice of electronic components and their placement in the CU which may offset the glider symmetry. Asymmetry caused by mass distribution is not considered a large problem however, since point masses can be used to compensate if needed. In addition, the REXUS program demands that the c.g. of the entire module going on the rocket is quite close to its cylinder axis.

The assumption about a calm atmosphere should also be considered. In normal flight the glider could encounter air pockets, wind and possibly bad weather. Even though the glider stability is somewhat tested this needs to be added to the simulation. Although weather is an important factor to consider for an aircraft, the Esrange team only launches rockets when the conditions for flight are optimal. That is, the simulation does not necessarily need to account for worst weather conditions possible because the glider would not be flown during it.

VII. CONCLUSIONS

This section presents the authors' conclusions based on the results presented in section V and the topics discussed in section VI.

The results in Fig. 6 - 9 are interpreted by the project members as having developed a functioning physics engine in the SIMULINK environment.

Because of this, the SIMULINK model described in section is believed to be a suitable platform in which a control system for autonomous steady gliding can be designed, given that the geometric design of the glider offers reasonable aerodynamic stability and that the weather conditions are assumed to be somewhat calm. More specifically the model is deemed a suitable platform for the development of control systems for gliding aircraft which uses an elevator/rudder design, and if other control surfaces are used the model needs to be re-worked.

The control system architecture presented in Fig. 5 works better for controlling the pitch and yaw angles than PID control, since the former has a faster performance and the PID yaw control even exhibit instability. In addition, the results in Fig. 16 and 17 implies that the performance of the proposed control system, from a stability standpoint, works over the entire spectrum of altitudes and reference commands that are considered in this project. However, due to the oscillations present in the transient it is concluded that tuning several control systems for different altitudes and utilizing a coupling logic to transition from one to the other would decrease the total energy consumption due to smoother control and as such allow for a longer flight time.

No consideration has been taken to the landing area such as objects blocking the flight path or existing no-flight zones. With that said, the project members agree that a landing sequence needs to be implemented to improve this part of the trajectory. Furthermore, it is concluded that the heading reference system is satisfactory and that the pitch reference can successfully be tracked without entering the stall region as discussed in section III-D3.

VIII. FUTURE WORK

There is still work to be done in the development of the flight control system before it is ready for actual flight testing and it is the aim of this section to shine light on future work needed and some of the proposed solutions from the authors toward this end.

A. Additional control

Since the focus of this project has been on the control of steady state gliding there still needs to exist some control system prior to the steady state flight and after Esrange is reached. The control system that is active before the steady state then needs to be in charge of deploying the wings and tail from the FFU and achieve stable flight conditions as the atmosphere becomes denser during the FFUs tumbling descent before switching to the control system developed in this project. Additionally, as mentioned in section VII, even for the steady gliding which is considered in this project, it is proposed that several controllers be tuned for different altitudes (and perhaps several data sets could be used from XFLR5 corresponding to different air densities) with some

switching mechanism between them as a function of altitude and/or velocity. It is also proposed that an additional flight control system be developed specifically for the landing sequence since it is more intricate than the gliding state. The switching between the glide state and the landing sequence could be implemented as a function of the altitude, velocity and distance to Esrange.

B. Saturate roll error instead of turn angle

At the time of writing this thesis, the turn angle $\Delta\Psi$ calculated in the heading reference system is saturated to achieve roll stability. That is, the turn angle is restricted to the interval of $\pm 30^\circ$ to ensure that the plane does not demand a too large rudder deflection, generating a too big roll angle. In theory however this saturation is usually placed on the roll error, which refers to the roll angle P controller input. Saturating this signal instead limits the roll angle directly which is the root of the problem. Doing this will also allow for greater turn angles which will probably yield a faster heading reference tracking during large turn angles. The method proposed of implementing this would be to remove the saturation in turn angle and tune the saturation in roll error for a 180° step in yaw until stability is achieved even when large turn angles are necessary.

C. Sensor fusion and filtering

The simulation environment is prepared for digital implementations of the sensors and a corresponding AHRS filter to interpret the sensor readings. The digital implementation of the sensors, the IMU block, is tune-able and was tuned in accordance with the data sheet corresponding to the on-board hardware. Though the project members were not able to tune the AHRS filter accordingly to obtain sufficient data to control the glider based on the data sheet tuned IMU model. This was partly due to confusions regarding data sheet and the block parameters of both the IMU block and the AHRS filter but the main reason was a simple lack of time. Even though the IMU/AHRS combination could not produce good results within the scope of this project, these blocks are deemed essential to keep and fine-tune for future development. In fact, if the controllers are fed faulty data from the AHRS filter the performance suffers greatly. The members of this project propose that the IMU and AHRS models be tuned according to actual sensor measurements instead of by the data sheet until the AHRS filter produces good results. This could be done by exporting only the AHRS model into C code, implement it on the microcontroller, get the AHRS input sensor readings via the FPGA to microcontroller UART line at 100 Hz, then save the sensor readings on a SD card along with AHRS output for later analyses and tuning. It is proposed that these tests be carried out both when the sensors are stationary, moving in space, and rotating during long time periods.

D. Virtual elevator/rudder to ruddervator coupling

The proposed way to go forward if a virtual elevator/rudder design is to be used, would be to tune the controllers as has been described in this thesis and use a coupling like one commonly used for RC plane radio controllers. A RC plane radio controller normally has a joystick for elevator/rudder commands and combinations in-between where if the plane has a V-tail, the elevator/rudder commands are coupled according to

$$\text{Right RV} = \text{Rudder} \cdot K_1 + \text{Elevator} \cdot K_2$$

$$\text{Left RV} = \text{Rudder} \cdot K_1 - \text{Elevator} \cdot K_2 \quad ,$$

where 'Right/Left RV' are the ruddervator commands and 'Elevator'/'Rudder' the elevator/rudder commands. The gains K_1 and K_2 then needs to be tuned after testing to achieve the desired result, where $K_1 = K_2 = 0.5$ would be the proposed initial test gains.

ACKNOWLEDGMENT

The authors would like to thank BSc groups I_1, I_2 and I_4, Nickolay Ivchenko, Christos Tolis and in particular Viktor Nan for all the late night troubleshooting.

REFERENCES

[1] Team PRIME, "PRIME Student Experiment Description", version 5-0. Internal KTH REXUS document, KTH, Stockholm, Sweden, Nov. 2019.
[2] Team B2D2, "B2D2 Student Experiment Description", version 3-1. Internal KTH REXUS document, KTH, Stockholm, Sweden, Jan. 2021.
[3] K. A. Kulbay and J. Nylöf, "High Altitude Glider Solution for Returning From Space," BSc thesis project I1, KTH, Stockholm, Sweden, May 2021.
[4] Chris Tolis, "The Data Hub PCB". Internal KTH REXUS document, KTH, Stockholm, Sweden, Feb. 2020.
[5] A. Malmberg and O. Munter, "Power and Electronics in Autonomous Glider for Sounding Rocket Experiments," BSc thesis project I4, KTH, Stockholm, Sweden, May 2021.
[6] XFLR5. (2021, May) XFLR5 web page. [Online]. Available: http://www.xflr5.tech/xflr5.htm
[7] D. A. Caughey. "Introduction to Aircraft Stability and Control", lecture notes in course "Introduction to Aircraft Stability and Control" (M&AE5070), Cornell University, New York, USA, 2011. [Online]. Available: https://courses.cit.cornell.edu/mae5070/Caughey_2011_04.pdf
[8] D. McLean, *Automatic Flight Control Systems*, 1st ed. 66 Wood Lane End, Heme1 Hempstead Hertfordshire HP2 4RG, UK: Prentice Hall International, 1990.
[9] V. Sazdovski, T. Kolemishevska-Gugulovska, and M. Stankovski, "Kalman filter implementation for unmanned aerial vehicles navigation developed within a graduate course," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 12–17, 2005, 16th IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667016382775
[10] M. Pedley and M. Stanley. (2021, Mar.) Basic kalman filter theory. USA. [Online]. Available: https://github.com/memsindustrygroup/Open-Source-Sensor-Fusion/tree/master/docs
[11] C. Veness. (2021, Mar.) Calculate distance, bearing and more between latitude/longitude points. UK. [Online]. Available: https://www.movable-type.co.uk/scripts/latlong.html
[12] MathWorks. (2021, Mar.) ISA atmosphere model. USA. [Online]. Available: https://se.mathworks.com/help/aeroblks/isaatmospheremodel.html
[13] Ublox. (2021, Feb.) MAX-8 Ublox 8 GNSS modules. [Online]. Available: https://www.tme.eu/Document/470e03d9c1a776f3fd1f142436381f21/MAX-8_DataSheet.pdf
[14] STMicroelectronics. (2021, Feb.) Mems motion sensor: three-axis digital output gyroscope data sheet. [Online]. Available: https://www.pololu.com/file/0J731/L3GD20H.pdf
[15] ——. (2021, Feb.) Ultra-compact high-performance ecompass module: ultra-low-power 3d accelerometer and 3d magnetometer data sheet. [Online]. Available: https://www.st.com/resource/en/datasheet/lsm303agr.pdf

[16] MathWorks. (2021, Mar.) AHRS - Orientation from accelerometer, gyroscope, and magnetometer readings. USA. [Online]. Available: https://se.mathworks.com/help/fusion/ref/ahrs.html?searchHighlight=ahrs&s_tid=srchtitle

Power and Electronics in Autonomous Glider for Sounding Rocket Experiments

Alexander Malmberg and Oskar Munter

***Abstract*—The aim of this bachelor's thesis is to design the electronics for an autonomous glider to be used in a sounding rocket experiment with return to launch site functionality. The electronics includes a battery solution, servos and a hardware platform for communication and control software. All of these parts need to be suited for a specified form factor and some extreme environments such as low temperature and vacuum. The electronics have been designed based on calculations for power consumption and temperature dependency. The system had to be power efficient since the space for batteries is limited. Servos were custom designed with motors and drivers to optimize both space and efficiency. Based on testing, simulations and calculations of the design the following can be concluded. The proposed system has the capability to meet the requirements to control and fly the glider all the way back to launch site even in a worst-case scenario. Thus an electronics system for the autonomous glider solution is feasible even with the strict requirements and conditions.**

***Sammanfattning*—Syftet med detta kandidatexamensarbete är att designa elektroniken till en autonom glidflygare vars uppgift är att återföra experiment uppskjutna med en sondraket. Elektroniken innefattar en batterilösning, servos för styrning samt en hårdvaruplattform för kommunikation och kontrollsystemet. Alla dessa delar ska implementeras i ett begränsat utrymme och klara av låga temperaturer samt vakuum. Beräkningar av energiförbrukning och temperaturberoende hos de olika komponenterna har gjorts för att designen ska klara förhållandena. Elsystemet måste vara effektivt nog för att kunna drivas med ett batteri kompatibelt med det givna utrymmet. Egendesignade servon är framtagna med motorer och drivare för optimerad effektivitet och storlek. Tester, simulationer och beräkningar visar att det föreslagna systemet är kapabelt att för de angivna kraven flyga glidflygaren tillbaka till basen. Elsystemet har även marginaler nog att klara detta under de mest påfrestande förutsättningarna.**

***Index Terms*—Servos for space, Low temperature batteries, Position sensor, REXUS/BEXUS,**

Supervisors: DR. Nickolay Ivchenko

TRITA number: TRITA-EECS-EX-2021:168

I. Introduction

Systems operating in space are becoming more crucial for today's society as they provide people around the world with services such as GPS, weather forecasts and land stewardship. Therefore, understanding the physical phenomena appearing and dictating the environment between earth and these systems is of importance. Because satellite orbits are not stable under a height of 300 km, only sounding rockets can be used to take confined measurements and perform experiments in the 50 to 200 km altitude range. In recent years student teams from KTH have participated in the REXUS program and conducted experiments and measurements with so-called free falling units (FFUs) that get ejected from sounding Rockets. Retrieval of the FFUs has been done by equipping them with parachutes and GPS receivers to enable search and pick up by helicopter. This however is costly and not always possible, risking the loss of research data. The next KTH student team applying for REXUS will try to solve this issue by making an FFU that turns into an autonomous glider capable of returning to a predetermined location. The first prototype design of this FFU is the goal of the bachelor thesis projects within context *I*. The focus of this report is the design of power and electronic system suitable for such a prototype. The system aims to tie together the signal and electrical power needed for the software control, wing deployment and movement of aerodynamic control suffices. Challenges for the system include withstanding the low pressure, high G-forces, vibrations and the high and low temperatures associated with the ascent of the rocket and the descent of the FFU. All while complying with the limited physical space, placement of mass requirements and the demands from REXUS.

Abbreviations

ADC	Analog to Digital Converter
BDU	Boom Deployment Unit
BDC	Brushed Direct Current
BLDC	BrushLess Direct Current
BOOMERANG	BOOM-deploying Experiment with Return-to-launch-site Automated Non-propelled Glider
BU	Base Unit
CU	Common Unit
FFU	Free Falling Unit
FPGA	Field Programmable Gate Array
LDO	Low Dropout Regulator
OCV	Open Circuit Voltage
PCB	Printed Circuit Board
PWM	Pulse Width Modulation
REXUS	Rocket EXperiment for University Students
RMU	Rocket Mounted Unit
SED	Student Experiment Documentation
SOC	State Of Charge
TOF	Time Of Flight

II. Background

A. BOOMERANG a REXUS Team

BOOMERANG is a team of KTH students aiming to participate in a future REXUS project. The team goal is to

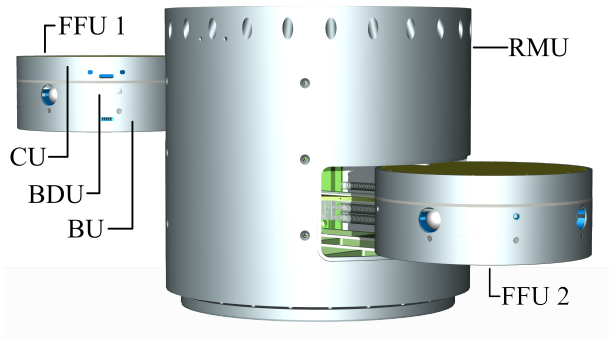


Fig. 1. Render of the RMU and FFU design from SPIDER-2 project

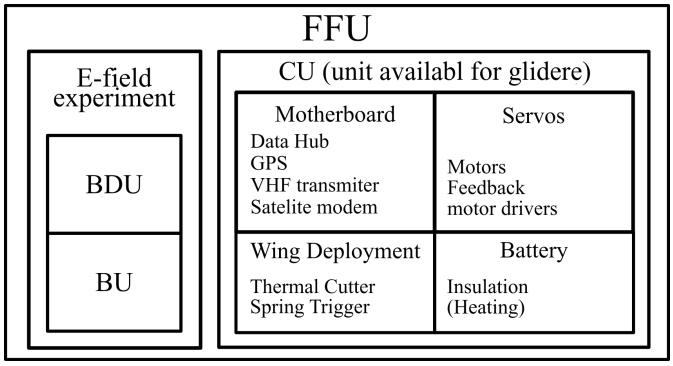


Fig. 2. Diagram of the CU

add an autonomous glide system to the FFU design for boom deploying electric field measuring probes first made by [1] and further developed by [2] for the KTH SPIDER project. The REXUS program is organized by Swedish National Space Agency (SNSA), German Aerospace Center (DLR), European Space Agency (ESA), Swedish Space Corporation (SSC) and Center of Applied Space Technology and Microgravity (ZARM). The program provides an experimental near-space platform for students of higher education via the use of single stage sounding rockets [3]. The program is what will enable the BOOMERANG team to execute the autonomously returning FFU experiment. That is if a so-called flight ticket is granted for a spot in a sounding rocket. The work done by the entire context including this report will contribute to a future application for this ticket.

B. Scope of Project I_4

In Fig. 1 the typical FFU design with the different units is shown, together with the RMU and an additional FFU. The scope of project I_4 is to design the electronics of the CU. The context groups were tasked with proposing an autonomous glider solution that would fit inside the CU of the FFU design originating from [1] and used in several KTH REXUS projects as well as KTH Space and Plasma Physics departments SPIDER projects. This way, tried and tested legacy designs for hardware, electronics and software can be reused. The Data Hub [4], microcontroller and FPGA card developed by the KTH PRIME REXUS team is a good example.

A big part of the CU electronics is the servos that make it possible to move the control surfaces of the glider to the right position. These servos had to be designed specifically for the task. This means that motors, drivers and position feedback all had to be investigated and decided on. Other parts of the electronics addressed are the power supply solution and the interface for signal processing and controlling.

III. Requirements and key Elements in the CU

In this section requirements and the key elements needed to fulfill the project objective are specified along with supporting theory. A block scheme of the different parts of the BOOMERANG system is shown in Fig. 2.

Many of the requirements are dictated by the fact that the system should be compatible and fit inside the CU. The space available for electronics is dependent on how much space is needed for the wings and wing deployment done by I_2. However, all components must have at least one outer dimension < 26 mm as it is the internal height of the CU. A hardware platform for implementation of I_3 control system as well as actuator for I_2 wing deployment is required. Capability of data transmission back to base and position tracking for flight path and retrieval. The servos for moving the control surfaces need to fulfill requirements specified for torque, speed and feedback on position. Required torque for moving the control surfaces is specified by I_1 to 0.01 Nm for typical flight and 0.1 Nm in a worst-case scenario [5]. The control system designed by I_3 requires the control surfaces to have sufficient speed of 50 rpm and a continues feedback on position with a accuracy of $\pm 0.5°$. The power source should have sufficient capacity to power the CU electronics from rocket ejection to landing, even in a worst-case scenario of max current draw and max TOF.

As BOOMERANG seeks to be part of the REXUS program the entire system aims to follow the restrictions and requirements of the REXUS User Manual [3]. This includes designing and picking components that theoretically should pass the tests in section 10 of [3].

A. Motors

To drive the servos mentioned in section II-B several motors could be used. The motors need to fit inside the given space of the CU and still be powerful enough to move the control surfaces. They also need to be efficient enough to be driven from a battery that also fits inside the CU. There are three possible motor types presented in this section.

BLDC motors have been used in previous REXUS experiments, such as B2D2 [6]. They have the advantages of small form factor and high speed. To control BLDC motors it is required to externally switch the current flowing through the motor. This means that the BLDC motor needs a more extensive control system than other options.

A common motor that is used in a wide range of applications where something needs to be moved to specific positions is the stepper motor. A stepper motor's position can be controlled without measuring the position with external sensors. This is beneficial since the control system only needs to provide signals for driving and does not have to receive any feedback

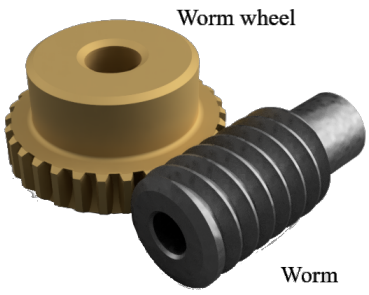


Fig. 3. Picture of the worm gear from the 3D render of the servos.

on the position. The position can not be fully trusted since the possibility of the motor skipping steps and rotate too far is likely. Thus the feedback could be flawed.

BDC motors are easy to drive since they require an ordinary DC voltage that only needs to be switched if the motor needs to change direction. BDC motors are also marginally bigger and not as fast as BLDC motors.

B. Reduction

Since it is possible to get high speed and hard to get any significant torque out of a small motor the servos need suitable gears for reduction. This will give the servos greater output torque in exchange for lower speed.

The reduction can be achieved with various types of gears and different number of stages. To calculate what reduction a specific pair of gears has, the number of teeth on the first gear is divided by the number of teeth on the second gear. The reduction is then used to calculate the output torque and speed for given input values. This is however only true for gears with zero losses. The efficiency of a gear is usually given as a percentage that tells how much of the input torque actually transferred.

It is also possible to construct a gear with different reductions when driven in a forward direction than in reverse. One type of gear that has this feature is the worm gear, see Fig. 3. It consists of one screw that is called worm and one gear that is called worm wheel. The worm is mounted with its axis angled 90° to the axis of the worm wheel. When the worm rotates with its threads between the worm wheels teeth it is able to move the wheel. The wheel rotates at a lower speed that is set by the reduction of the worm gear. The wheel however is not able to move the threads of the worm and this means that the gear is locked from any reverse drive. The worm gear has a lower efficiency when compared to other gears. This is much due to the friction between teeth and threads that follows from [7].

C. Drivers

To be able to drive a BDC motor in both directions and with varying speed, an additional driver is required. The driver needs to be able to control the value and direction of the current flowing through the motor. BDC motor drivers enable this functionality with the use of an H-bridge. The H-bridge can be implemented with different types of transistors, the most basic one having P-MOSFETs on the high side and N-MOSFETs on the low side. That implementation is easy to switch since both sides can be controlled with the same voltage level.

However, there are more efficient implementations of an H-bridge. P-MOSFETs have a higher on-resistance than N-MOSFETs. This is because the charge carriers in N-MOSFETs, electrons, have much higher mobility then those for P-MOSFETs, holes [8]. It is therefore preferable to only use N-MOSFET. The usage of N-MOSFET for both high and low sides of the H-bridge does come with some complications as well since the high side would need a gate voltage much higher than the drain voltage. This is often solved with an additional gate driver that can deliver the right threshold voltage.

D. Control Surface Feedback

A minimal delay on the feedback of control surface angel is needed to implement a proper closed control loop for autonomous steering. Encoders or position sensors for the control surfaces are therefore needed. These can either be absolute or incremental.

An incremental encoder added to the motor axis is often available with BDC motors and can provide the angle feedback if the gear reduction is accounted for. This approach introduces an error caused by the backlash in the gearing if the motor shaft angel is simply divided by the gearing reduction. Having the encoder on the same rotating axis as the surface instead of directly on the motor shaft neglects the error caused by backlash in the gear reduction.

Measuring position with incremental encoders is done by counting pulses from the output signal. Each pulse represents a rotation with a specific angle, which varies with the encoder's resolution. The signal is typically sent over two channels with identical pulses, only with different offsets. This makes it possible to determine the direction of the rotation. The benefit of incremental encoders is that they can be very precise without taking up too much space.

An absolute encoder will directly provide the position of a rotating shaft. This is beneficial since there is no need for software to keep track of and remember the position. The most basic absolute encoder can be built with a potentiometer. This will provide an analog output signal with a voltage that varies with the position. But there are important considerations that come with a passive component since its material characteristics can affect its behavior.

E. Batteries

As the CU is electrically isolated from the rest of the FFU it will need its own power source to power any system inside, after the ejection of the FFU from the RMU. A battery solution is therefore needed. When choosing a battery for a sounding rocket the harsh conditions need to be accounted for. The Low pressures of sub 0.5 mBar specified in section 10 of [3] and sensor data from appendix B-A indicating temperatures inside FFU:s below -20 °C are the two most limiting factors. High energy density (Wh/kg) and volumetric energy density

(Wh/L) is preferable as space in the CU is sparse and keeping mass down helps the gliding capability. Recharge capability is preferable for ease of use. The readily available rechargeable battery types with the highest energy density volumetric and by mass are Lithium based according to [9], Lithium-ion (Li-ion) and lithium-ion polymer (LiPo) being the two most common ones. In [10] it is evaluated how well suited different battery types are for use in satellites and Li-ion are shown to have the highest Wh/kg and Wh/L out of all considered rechargeable batteries. The use of LiPo batteries is allowed but discouraged in [3] due to their delicate discharge and mechanical sensitivity.

There exist different types of Li-ion batteries but typically they are constructed in cells that can hold different capacity mainly dependent on size. However, independent of capacity size Li-ion cells typically have a nominal voltage of 3.7 V and maximum voltage when fully charge at 4.2 V. Li-ion batteries performances regarding capacity and discharge rate are heavily affected by temperature especially below 0 °C as shown by [11] and [12]. In [11] it is experimentally shown how the capacity of a Li-ion battery drops with 31 % at -30 °C compared to capacity at 0 °C, even at a relatively low discharge rate of $C/20$. $1C$ is the current to discharge the battery in one hour. It is also stated that the capacity drop gets increasingly nonlinear at very low temperatures, sub -25 °C. The performance drop is derived to be caused by increasing internal resistance R_i with lower temperatures. The increase in R_i causes the OCV to drop significantly reaching critically low voltages, even at high SOC. This effect needs to be accounted for as the supply voltage for the Data Hub is 3.1 V to 4.5 V [4]. The Data Hub needs to be powered on for the entire experiment. Hence OCV of the battery may under no circumstances drop below 3.2 V.

F. Thermal Management

The BOOMERANG FFU will be exposed to a large temperature span and quickly changing ambient conditions. Some of the outer parts of the rocket can reach up to 110 °C within 50 s after launch according to [3]. This heat will radiate into the FFU. After ejection, the FFU is instead exposed to the ambient temperature of the atmosphere. The temperature in the atmosphere varies greatly but typically reachs as low as -56 °C at altitudes of 10 km to 20 km above sea level according to the US Standard Atmosphere shown in [13]. Data from temperature sensors on SPIDER-2 FFU:s can be found in appendix B-A and shows how the inside of FFU:s cools down during the descent as shown in Fig. 8. Time of flight for the gliding part of the descent, starting at 10 km to 20 km can reach up to 90 min according to simulations by [5] and [14]. Lower temperatures than the ones recorded by SPIDER-2 FFU:s are therefore to be expected inside the BOOMERANG FFU as it will spend more time in the coldest part of the atmosphere.

These conditions need to be accounted for during the design of the system. Using tried and tested designs and components exposed to similar conditions such as the ones used in previous KTH REXUS and SPIDER projects can greatly reduce the extensive testing needed to confirm reliability.

Thermally isolating more temperature-sensitive components such as batteries is an option. Heat flow q perpendicular through a material with thermal conductivity constant k, area A, time span Δt and temperature difference of ΔT from one side to the other is given by Fourier's law of heat conduction,

$$\frac{q}{\Delta t} = k\frac{A\Delta T}{L}. \tag{1}$$

Insulation capability therefore increases linearly with the thickness L of the material given this simple static scenario. How change in heat energy E_h of a mass m affects its temperature can be described by

$$\Delta E_h = mC_h\Delta T, \tag{2}$$

where C_h is the specific heat capacity of the mass.

IV. Implementation of the CU Electronics

In this section, it is described how all the contents of the system were implemented. The different design choices will be explained with and based on the theory of the different contents in section II-B and requirements stated in section III.

A. Motors

The chosen motor for this project is the Faulhaber 1024 006 SR. It is a BDC motor with a nominal voltage of 6 V and nominal speed of 7460 rpm [15]. The reason for choosing BDC motors over other alternatives was mainly due to the much simpler driving and control. With BLDC motors it would have been required to add an extra FPGA to the system since the currently used Data Hub does not have enough pins for controlling two of those motors. Stepper motors were not chosen because there were no identified benefits beyond being able to control the position. It is not necessary to control the position of the shaft on the motor since it will not give any accurate information about the control surface. This is because gears and other mechanical couplings could give some delay and play to this motion.

Several BDC motors were investigated. All of them were plotted with speed against torque in MATLAB, see Fig. 4 for the chosen motor. This method made it clear which motors were suitable and gave a good overview of how size, voltage and current affected each other. The chosen motor from Faulhaber showed to have the lowest rpm-torque curve gradient and an associating low current draw, making it the most suitable option. The speed of the motor was also sufficient for the control surfaces.

B. Motor Driver Boards

A PCB was designed for the two identical motor driver circuits to be assembled on. One finished board can be seen in Fig. 5. In the middle of the board, there is a hole for the output axis to pass through and this is so that the output angle can be measured.

To drive the motors, implementation of an H-bridge was done with the DRV8850 chip from Texas Instruments [16]. It is a 5.5×3.5 mm chip with a powerful H-bridge consisting

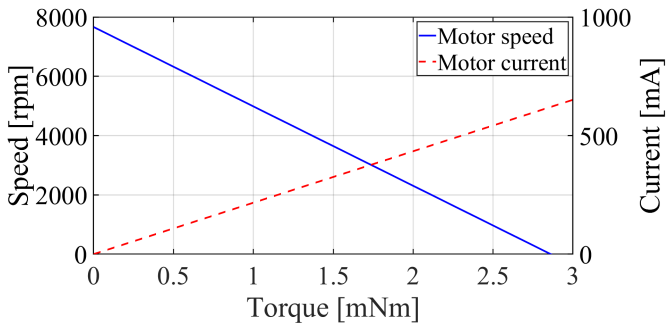


Fig. 4. Speed and current plotted for different torque applied to the output shaft of the Faulhaber 1024 006 SR when driven at 3.7 V, values taken from datasheet [15].

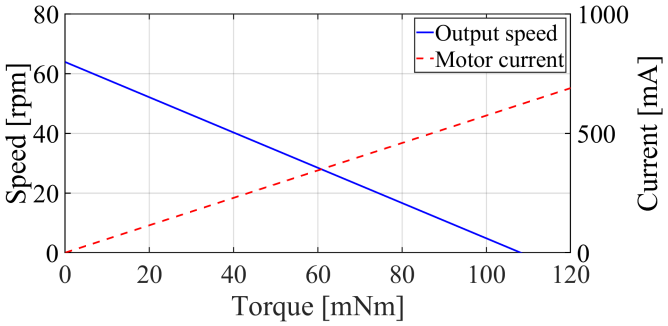


Fig. 6. Speed and current plotted for different torque applied to the output shaft of the gears when 3.7 V is applied to the motor and the gear efficiency is 31 %, values taken from Fig. 4 and datasheet [18].

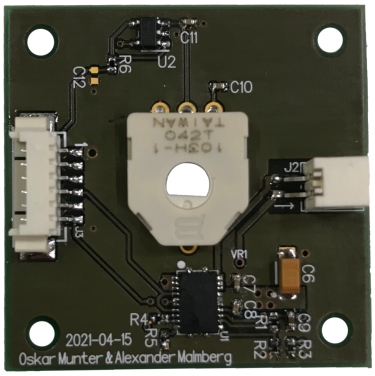


Fig. 5. Completed motor driver and feedback PCB

of four N-MOSFETs and integrated gate drivers. The on-resistance is 65 $m\Omega$ which results in low losses. It also includes an LDO that can be used as a stable voltage supply for other systems to be used with the driver.

The sensor for measuring the output angle was implemented with a potentiometer. The component of choice was the Bourns 3382. This potentiometer has a very small form factor and has precise linearity, better than ± 2 % [17]. The linearity was also tested manually and it turned out to be even more precise.

C. Motherboard

The motherboard has a wide range of functionalities where the Data Hub will serve as the main unit for control and managing data. These functionalities are an autonomous flight control system, a wing deployment system, a position transmission system and the possibility of a battery heating control system. The schematics for the motherboard and the required components are specified in appendix A-A.

Most of the designs for the systems on the motherboard are inherited from earlier rocket experiments. This is because they have been tested several times and proven to be reliable. The layout for the motherboard is not yet done since the interior of the CU and the thermal management is not fully decided on.

A layout for the autonomous flight control system was designed and milled on a copper board. This was done to be able to test the control system's functions. The circuit board has a Data Hub and a GPS board attached to it including an interface between them and the motor driver boards.

The wing deployment control system will be implemented with a thermal cutter that has been used as an actuator in many previous sounding rockets. It will burn off a thin nylon line that is holding the spring-loaded wings inside the FFU. The heat is created by letting a high current flow through a kanthal wire helix that encloses the nylon line. This helix will be mounted directly on the motherboard between two bolts.

The position transmission system consists of both a radio transmitter and a satellite modem for transmitting over the Globalstar satellite network. This means the position can be transmitted through two channels independent from each other.

D. Gears

The reduction between the motor and the output shaft to the control surfaces was implemented with two stages of gears. In total the reduction ads up to $120:1$. The first stage is a gear head directly attached to the motor and has a reduction of $4:1$. The second stage is a worm gear with the reduction $30:1$.

The efficiency of the gear head is 90 % according to [18] and the worm gear has an efficiency between 30 to 40 %. The efficiency of the worm gear is hard to specify since it depends on many different parameters, such as the mounting of the shafts holding the gears. The estimated efficiency was given by the company that provided the worm gear, Mekanex. If these two efficiencies are combined the result is approximately 31 %. With this value, the output speed and torque of the worm gear can be calculated, see Fig. 6. The chosen motor with this gear setup has sufficient torque for moving the control surfaces specified by I_1 [5].

For the prototype, the gears were mounted in a 3D printed block, see Fig. 7, and tested together with the motor as described in section V.

E. Battery

The proposed battery to power the system is the SAFT MP 174865 xlr, a rechargeable Li-ion one cell battery with a capacity of 5.3 Ah [19]. More precisely a Li-ion cell of Nickel Manganese Cobalt Oxide type. The SAFT MP series batteries have been used in previous KTH REXUS projects as well as

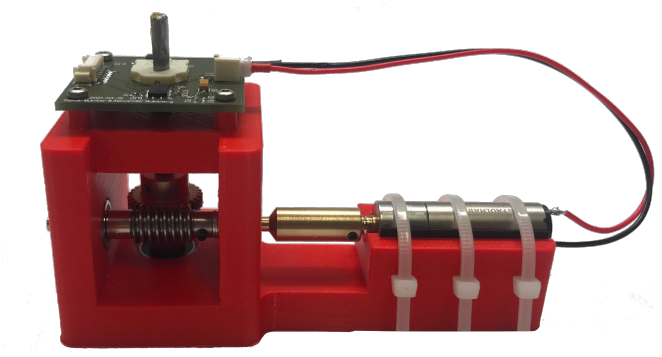


Fig. 7. Test setup with driver and feedback PCB, gears and motor mounted together in a 3D printed block

TABLE I
TABLE OF POWER CONSUMPTION

Component	Typical [mA]	Max [mA]	Duration	mAh
2×Motors	260	1200	90 min	390
2×Feedback	2	-	90 min	3
Globalstar [20]	30	350	100 min	50
TX1 [21]	11	-	100 min	18.5
Data Hub	200	-	100 min	333.5
GPS board	50	-	100 min	83.5
Thermal cutter	3000	-	10 sec	10
Total (Typical)				888.5
Total (Motors at stall current)				2498.5

in the SPIDER-2 FFU:s. Legacy RMU and FFU electronic designs are therefore compatible with this battery making the ease of integration to the modular CU design very high. SAFT Li-ion is the recommended rechargeable batteries by REXUS as stated in section 7 of [3]. The possibility to use more easily available 18650 Li-ion batteries was investigated. However, the ease of integration, REXUS recommendation and the known capability of use in FFU:s favored the SAFT MP battery.

As stated in III-E the Data Hub requires a supply voltage of 3.1 V to 4.5 V. The Data Hub needs to be powered on for the entire experiment. It is therefore crucial that the OCV of the battery never drops below 3.2 V. The MP 174865 xlr reaches a potential of 3.2 V when 5.1 Ah or 4.85 Ah have been discharged at a discharge rate of 1.06 A ($C/5$) respectively 2.65 ($C/2$) at 20 °C [19].

To assess what the system demands from its power source in this case, a simple current draw and Ah calculation was done by summing up all of the component's current draw to receive the discharge rate of the battery for different parts of the flight. Adding the total time each current is drawn will give the total capacity needed. Both of these calculations are done twice, a worst-case and a nominal scenario. The result of the calculations and each component current draw is shown in Tab. I. The values have been received by measurement of available components, specifications in datasheets and available documentation from previous KTH REXUS and SPIDER projects.

Mention in section III-E Li-ion cells OCV significantly drops when operating in low temperatures. As stated in section III-F the temperature inside the FFU and thus also the battery will likely be below −20 °C. Exactly how low the temperature would be for a max TOF scenario is not known. The MP 174865 performance at different low temperatures is presented in [19] and indicates it reaching the critical minimum supply voltage for the electronics of 3.2 V when 4.6 Ah, 3.65 Ah or 2.4 Ah have been discharged at 0 °C, −20 °C respectively −30 °C at a discharge rate of 1.06 A ($C/5$). The stated minimum operating temperature is −35 °C and performance below is not stated.

F. Thermal Insulation

As mentioned in section III-F temperatures below −20 °C are to be expected inside the FFU. With such low battery temperatures, the reduced amount of battery capacity available before an OCV of 3.2 V is reached as shown in section IV-E could cause a critical failure of the system. This is if current draw equal to the worst-case scenario calculated in Tab. I occur. No matter the discharge rate, if its temperature drops below −35 °C the performance gets exponentially worse due to increasing R_i as described in Section III-E. It could even malfunction completely as it is only rated to −35 °C. To save performance and prevent a complete malfunction the battery should have a temperature at least above −30 °C during the entire experiment.

Keeping the battery warm could be done by insulating it. Insulating the battery would however take up valuable space inside the CU. To get a better understanding of the relationship between insulation thickness and battery temperature a simple battery temperature simulation was made in Matlab. The simulation approximates battery temperature through an iterative process. First calculating heat flow through insulation of L thickness and the battery surface area at time t, using (1). Then with (2) the new battery temperature for the next iteration $t + 1$, can be set.

Using this method simulations of battery temperature for the MP 174865 were made. For all simulation battery heat capacity was set to $C_h = 950$, a typical number for Li-ion batteries according to [22]. Insulation was programmed with thermal conductivity $k = 0.3$ typical for styrofoam and varying thickness L. To mimic the varying descent temperature the SPIDER-2 temperature data was used as the cold side temp of the insulation. For a worst-case scenario, the temperature was set to a constant −40 °C. The most relevant results are shown in Fig. 8 and 9. Adding different amounts of heat energy inside the insulation was also tested and is shown in the same figures.

V. Testing and Results

A setup for one servo with motor and finished driver PCB was built to test their functionality together with a prototype motherboard, same setup as shown in Fig. 7. The prototype motherboard was equipped with a Data Hub for testing. This way motor driver capability and feedback performance tests could be performed. As the Data Hub can send control signals and read the feedback signal from the potentiometer with the built-in ADC. The ADC data can be saved at a high sample

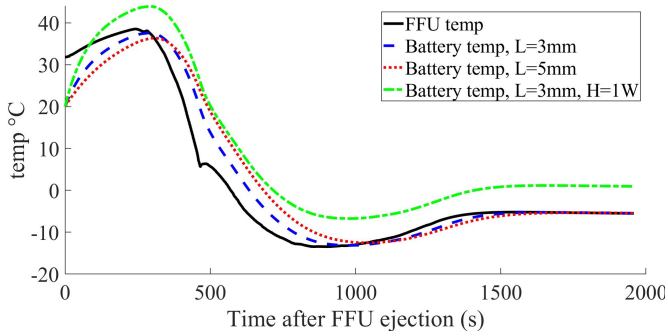


Fig. 8. Results from battery temperature simulation using SPIDER-2 temperature data as baseline. L referring to insulation thickness and H to added heating in watts.

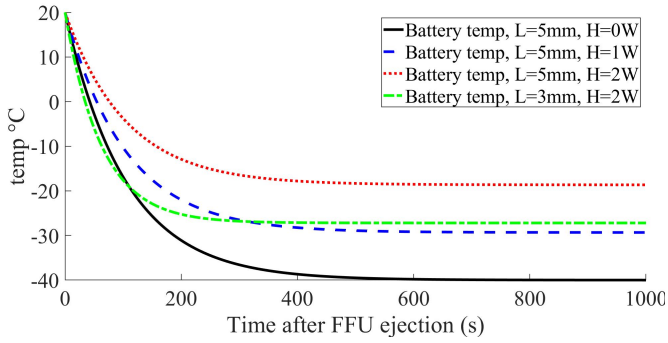


Fig. 9. Results from battery temperature simulation with inside FFU temperature set to a constant of -40 C. L referring to insulation thickness and H to added heating in watts.

rate for accurate plotting afterward. The testing confirmed that the driver could drive the motor in either direction using only two output pins, both at battery voltage and at varying speed using PWM. From Fig. 10 the time for one full rotation of the output shaft can be read and the rotation speed can be derived to 54.5 rpm.

Tests using simple P controller code demonstrate the servos capability to accurately move to a given angle as shown in Fig. 11. Angle accuracy can be shown with feedback voltage as it is directly correlated to the angle of the output shaft. The test shows that the feedback on the output shaft angel given by the potentiometer is accurate but not sufficient to achieve the required precision. However, this is due to noise in the Data Hub ADC exceeding the accuracy of the potentiometer feedback.

VI. Discussion

A. Battery

The choice of a 5.3 Ah battery when power consumption shown in Tab. I indicates only 888.5 mAh is needed for the typical case of a 90 min TOF scenario may seem odd or even flawed as it adds unnecessary weight and take up a large space in the CU. The performance drop of Li-ion battery stated in section III-E and how it affect the proposed electrical system in section IV-E indicates that an oversized battery is needed in the application. Exactly how the MP 174865 xlr OCV is affected by current draw peeks larger than 1.06 A at sub-zero

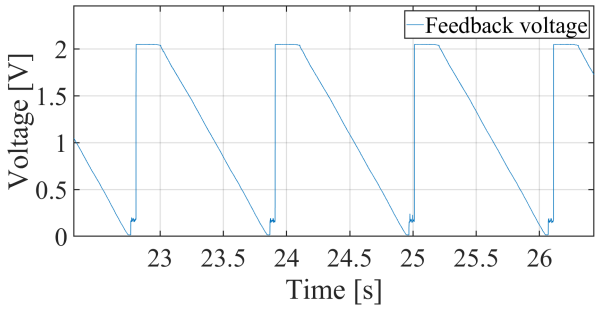


Fig. 10. Voltage on feedback signal from potentiometer when the motor is driven at 3.7 V continues in one direction.

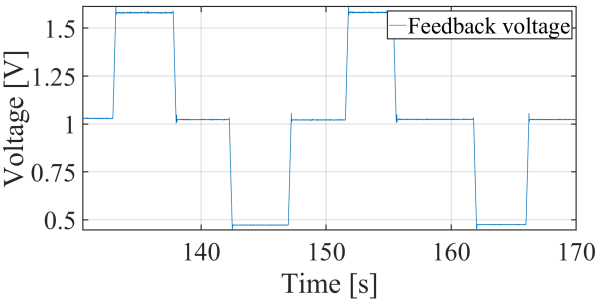


Fig. 11. Voltage on feedback signal from potentiometer when the motor is driven at 3.7 V and the output angle is controlled to 90 °, 0 ° and −90 °.

temperatures which are highly likely is not known. One current draw peak causing an OCV drop below 3.2 V is all it takes for the system to fail. A large safety margin was therefore chosen.

The current draw in a worst-case scenario shown in Tab. I may seem extremely unlikely, and it is in a normal flight scenario. However, in a mechanical breakdown scenario it is not. To have sufficient capacity for such a scenario is desirable as the FFU could continue to transmit its location and potentially be found. This way the E-field experiment data could possibly be saved even if the glider fails.

B. Insulating and Heating the Battery

As shown in section IV-F adding even a thin layer of insulation material make a difference to the battery temperature. With the chosen battery being rated for a minimum temperature of -35 °C and the risk of this happening during a max TOF scenario is considered high, hence some sort of insulation is considered needed. Fig. 9 indicates that even with relatively low heating of 1 or 2 Watts an equilibrium of dissipating and added heat can be reached 15 to 25 °C above the ambient temperature inside the FFU. Using capacity of the battery to keep itself from going to cold may be necessary. This could possibly even extend the capacity discharge range before the critical OCV of 3.2 is reached. This is supported by the fact that OVC of Li-ion batteries drops exponentially at temperatures below -25 °C as stated in section III-E and the capacity cost of heating is linear. However a more sophisticated thermal analysis of the battery, insulation and crucially a better approximation of the temperature of the FFU inside for a max TOF from 20 km altitude is needed.

Consequentially a definitive answer on how thick insulation and how much if any heating is needed can not be given.

C. Reduction Efficiency

The designed servos are considered to have a sufficiently low power consumption for the given task. However, the efficiency of the chosen worm gear is low. There are other components of choice for the worm gear that could lower losses in the reduction. This would further reduce the battery capacity needed for an entire flight. The efficiency achieved with other components will still be lower than for example planetary gears but the reverse lock functionality was considered absolutely necessary.

D. Motor Control

Test and associated results presented in section V indicate that the servo has sufficient speed for the application although this can not be said with complete certainty. This is because of the speed dependency on the torque load shown in section IV-A. The low torque of 0.01 Nm during a typical flight is however most likely not far from the load during the test making the results still relevant.

The required accuracy was not reached in the P controller test. This is due to noise in the Data Hub ADC exceeding the accuracy of the potentiometer feedback. Achieving the required accuracy of $\pm 0.5^\circ$ is therefore still possible with the current feedback hardware. The feedback signal can for example be filtered or averaged out over a few samples. An even better solution would be to drive the motors with PWM and implement a proper closed loop PID controller. These improvements in software would also eliminate the overshot visible in Fig. 11 and any static offset often found with simple P-controlled systems.

E. Future Work

Work on the project can continue by optimizing the software control of the motors as discussed in section VI-D to confirm that the ± 0.5 ° accuracy requirement can be achieved. A better thermal analysis is needed to confirm the exact thickness of insulation for the battery and further investigate heating possibilities. The layout, construction and testing of the motherboard design is future work necessary to confirm the compatibility of all the different parts of the system. If a flight ticket is granted for the REXUS 31/32, testing new components and the entire system according to section 10 in [3] will have to be done.

VII. Conclusion

The proposed electronics and power system for the CU of the BOOMERANG FFU is summarized in Tab. II. The components of the motherboard have all been tried and tested in previous sounding rockets and therefore fulfill the REXUS-requirements. The same goes for the SAFT battery, it also fulfills the capacity requirement as shown in section IV-E under the circumstance that it is kept above -30 °C. To ensure this we have concluded that some sort of battery insulation is needed and possibly heating, exactly how this should be done needs further investigation. The results from testing done of the servo indicate that the hardware is sufficient for the task and can meet the speed, strength and accuracy requirements with optimization through software. All components in the servos are rated to handle the harsh conditions and fit in the CU. However, some further testing to verify this will have to be done in the future. If the space allocation between I_4 and I_2, max TOF, or any other key parameter does not drastically change the following can be concluded. A system such as the one suggested and partly prototyped in this report has the capability to support all the other context I projects. Motivated by the fact that it has the capability to meet all the requirements necessary. A successfully autonomous glide back to launch site is therefore highly feasible from a power and electronics standpoint.

TABLE II
HARDWARE COMPONENTS OF THE SYSTEM, COMPONENTS IN PARENTHESES ARE SUGGESTED BUT NOT CONFIRMED

Component	Description
Motherboard	
Data Hub	Programmable CU & FPGA
GPS	GPS receiver and supporting electronics
Globalstar	Satellite modem
Thermal cutter circuitry	Electronically controlled actuator
TX1	VHF radio transmitter
Umbilical circuitry	Interface to RMU and ejection detection
(Heating circuitry)	Circuitry to control heating of battery
Servo	
Faulhaber 1024 006SR	BDC motor
Faulhaber gearbox	Planetary gear head 4:1 reduction
Worm gear	Mechanical one way gear 30:1 reduction
PCB	Electronics platform
Motor driver	H-bride and with integrated gate drivers
Absolute feedback	LDO and potentiometer driven angel encoder
Power Supply	
SAFT MP174865 xlr	Rechargeable battery
Insulation	Thermal insulation of battery
(Heating element)	Electrical heating for battery

Appendix A
Schematics

A. Motherboard

B. Motor driver board

Appendix B
Spider 2 Data

A. Temperature inside the FFUs

Acknowledgment

The authors would like to thank supervisor Dr. Nickolay Ivchenko for his support and engagement in the project as he truly exceeded on what he was obligated to do and inspired us. Christos Tolis for guidance regarding PCB design and valuable insight in previous KTH REXUS and sounding rocket work. The entire BOOMERANG Team for all striving to reach the common goal of a successful REXUS application. The Brutal Brewing company for their steady hand soldering solution.

REFERENCES

[1] N. Ivchenko, G. Olentsenko, G. Tibert, and Y. Yuan, "Isaac: A Rexus student experiment to demonstrate an ejection system with predefined direction," in *Proceedings of the 22nd ESA Symposium on European Rocket and Balloon Programmes and Related Research*, Jun 2015, pp. 235–242.

[2] J. Asplund, "Design and implementation of a sounding-rocket electric-field instrument," Master's thesis, KTH Royal Institute of Technology, Stockholm, Sep. 2016.

[3] K. Schüttauf. (2018, Oct.) Rexus user manual. [Online]. Available: http://rexusbexus.net/rexus/rexus-user-manual/

[4] C. Tolis, "The data hub pcb," KTH Royal Institute of Technology, Stockholm, Tech. Rep., Feb. 2020.

[5] K. A. Kulbay and J. Nylöf, "High altitude glider solution for returning from space," BSc. thesis, KTH Royal Institute of Technology, Stockholm, May 2021.

[6] A. Jansson and A. Lezdins, "Student rocket experiment b2d2 - power system," BSc. thesis, KTH Royal Institute of Technology, Stockholm, Jul. 2020.

[7] B. Magyar and B. Sauer, "Calculation of the efficiency of worm gear drives," in *International Gear Conference 2014*. Lyon Villeurbanne, France: Elsevier Science, Chandos Publishing, 2014, pp. 15–23.

[8] H. Chenming, *Modern Semiconductor Devices for Integrated Circuits*. New Jersey: Pearson/Prentice Hall, 2009, ch. 6.

[9] M. M. Thackeray, C. Wolverton, and E. D. Isaacs, "Electrical energy storage for transportation—approaching the limits of, and going beyond, lithium-ion batteries," *Energy Environ. Sci.*, vol. 5, pp. 7854–7863, 2012. [Online]. Available: http://dx.doi.org/10.1039/C2EE21892E

[10] G. Rao and R. Pandipati, "Applications – transportation — satellites: Batteries," in *Encyclopedia of Electrochemical Power Sources*, J. Garche, Ed. Amsterdam: Elsevier, 2009, pp. 323–337. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780444527455003786

[11] A. A. Hussein, "Experimental modeling and analysis of lithium-ion battery temperature dependence," in *2015 IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2015, pp. 1084–1088.

[12] L. Zheng, J. Zhu, G. Wang, D. D.-C. Lu, P. McLean, and T. He, "Experimental analysis and modeling of temperature dependence of lithium-ion battery direct current resistance for power capability prediction," in *2017 20th International Conference on Electrical Machines and Systems (ICEMS)*, 2017, pp. 1–4.

[13] W. Vaughan, "Standard atmosphere," in *Encyclopedia of Atmospheric Sciences*, J. R. Holton, Ed. Oxford: Academic Press, 2003, pp. 2107–2113. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B0122270908003791

[14] M. Schmekel and L. Ringaby, "Simulation and control system design for autonomous gliding to a given location," BSc. thesis, KTH Royal Institute of Technology, Stockholm, May 2021.

[15] Faulhaber, *DC-Micromotors Series 1024 SR*, Croglio Switzerland, Feb. 2020, 1024-006-SR Datasheet. [Online]. Available: https://www.faulhaber.com/en/products/series/1024sr/

[16] Texas Instruments, *DRV8850 Low-Voltage H-Bridge IC With LDO Voltage Regulator*, Nov. 2013, DRV8850RGYR Datasheet. [Online]. Available: https://www.ti.com/product/DRV8850

[17] Bourns, *3382 - 12 mm Rotary Position Sensor*, Mar. 2017, 3382H-1-103 Datasheet. [Online]. Available: https://www.bourns.com/products/sensors/position-sensors/rotary-position-sensors-aec-q200-compliant/product/3382

[18] Faulhaber, *Planetary Gearheads Series 10/1*, Croglio Switzerland, Feb. 2020, 10/1-4:1 Datasheet. [Online]. Available: https://www.faulhaber.com/en/products/series/101/

[19] Saft, *3.65 V high energy Li -ion cell with robust performance and cycle life*, Levallois-Perret France, Apr. 2018, MP 174865 xlr Datasheet. [Online]. Available: https://www.saftbatteries.com/products-solutions/products/mp-small-vl

[20] Globalstar. (2021, Apr.) Stx3 satellite transmitter. STX3 Datasheet. [Online]. Available: https://www.globalstar.com/en-us/products/commercial-iot/stx3

[21] Radiometrix, *VHF Narrow Band FM Transmitter and Receiver*, Aug. 2012, TX1 Datasheet. [Online]. Available: https://www.radiometrix.com/content/tx1

[22] K. K. Parsons, "Design and simulation of passive thermal management system for lithium-ion battery packs on an unmanned ground vehicle," Master's thesis, California Polytechnic State University, San Luis Obispo, Dec. 2012. [Online]. Available: https://digitalcommons.calpoly.edu/theses/912/

CONTEXT J – PART I
DESIGN AND TESTING OF NOVEL MICROWAVE & ANTENNA TECHNOLOGIES

POPULAR DESCRIPTION

Connect to anything, anywhere, anytime

Imagine a world where cars, houses, and even fridges could talk. With new antenna technologies and the rollout of 5G, it will become possible to connect to anything anywhere, faster than ever before. The new challenge for the telecom industry is connecting the growing number of devices with novel antenna designs.

5G is the fifth and newest generation of mobile networks. Mobile networks are what makes it possible to wirelessly connect to the internet on your phone without being connected to WiFi. They work by transferring information, for example, an image or a video, inside radio waves, which are sent out from antennas. These radio waves can have different frequencies, meaning that they oscillate at different speeds. One thing that makes 5G different from previous generations of mobile networks is the use of higher frequencies.

Using higher frequencies in 5G includes faster and more reliable internet, for the common user and new opportunities for the industry. Higher frequencies allow for smaller antennas that require less material to manufacture and that can come in new forms.

Lens antennas are one of the novel antenna technologies that could be viable for the telecom industry. These antennas are large at lower frequencies but decrease in size with an increase in frequency. Lens antennas work similarly to optical lenses, such as regular glasses, in the way that they can direct and focus electromagnetic waves. In new designs studied at KTH Royal Institute of Technology, lens antennas can easily and cheaply be manufactured and combined with conventional antennas.

Since higher frequencies propagate a shorter distance, a higher density of base stations will be needed. This coupled with smaller antennas will lead to a larger capacity to support more phones and smart appliances. This would allow for everyday objects to be able to communicate with each other over the Internet, giving room for talking cars, houses, and other appliances. This is called the Internet of Things or IoT.

Going from 4G to 5G and eventually, an even faster generation, 6G, brings forward new opportunities for an increasingly digital world. There is no visible end to technological evolution and although 5G and IoT are two of the hottest tech topics today, they will soon be as outdated as carrier pigeons.

SUMMARY OF PROJECT RESULTS

With a new space era at our doorstep, the need for antenna technology has never been greater. This has led to the research of antennas and filters that may be used in spacecrafts to make sure high quality data transmission can be made, while still being compact and reliable. Because of these antennas and filters consisting of simpler structures, more precise and low cost manufacturing can be made.

One alternative to selecting a high quality antenna while still being compact is the Lens antenna. Lens antennas can be used in combination with conventional antennas to increase the directivity at a low cost. Lens antennas do not require a complex feeding network and the antenna size can be made small in applications for higher frequencies. This makes them an interesting solution for future communication systems.

The project groups in J1 have designed and manufactured Luneburg lens antennas in the X-band, 8-12 GHz. The Luneburg lens creates a plane wave at the antipode when excited with a cylindrical wave at the periphery of the lens. The Luneburg lens can be realized with a dielectric using spherically symmetric gradient index or by creating a geodesic lens with air filled parallel-plate waveguides. Using a geodesic lens eliminates the need for using dielectrics, which introduces losses to the antenna. Without the need for complex dielectrics, geodesic Luneburg lenses can be cheaply manufactured in plastic with 3D-printers, and then coated with a thin film of metal.

Project group J1a has designed and manufactured a modulated geodesic Luneburg lens with the goal of minimizing the overall height of the antenna. The height is reduced by folding the lens, creating a new lens with the same focusing properties. The lens is fed from a microstrip antenna into a horn antenna leading to the lens. To achieve the required bandwidth for the microstrip antenna various bandwidth increasing techniques were employed. In further research projects, alternative feeding methods and mechanics to move the port radially may be explored.

Project group J1b has designed and manufactured a geodesic reflecting Luneburg lens. The reflective Luneburg lens has the same characteristics and advantages as the Luneburg lens, but gets a larger aperture by reflecting the wave to the outside of an upper waveguide. This is achieved by using a different refractive index profile between two parallel plates to get the desired ray path. The lens has been modulated in order to decrease the overall height and is fed with a waveguide connected to a coaxial cable . In further research projects it could be tried to design a geodesic reflective Luneburg lens that can reflect into an electric band gap (EBG) structure to create a leaky wave antenna with pins in order to increase directivity. The lens could also be manufactured for higher frequencies, making the lens smaller overall.

Waveguide filters have been an alternative to filter radio signals for a long time, however these structures have mostly relied on complicated structures such as beds of nails and ridges. The glide symmetric holey electric band gap (EBG) technology is a good choice for simplistic structures as well as good filtering abilities.

Project group J2 has designed and manufactured a 5G/6G bandpass filter in the Ka-band using periodic structures, more precisely glide-symmetric holey EBG structures. These structures rely solely on holes and their placement to create a wide stopband which then can be used to create a bandpass filter. Moreover, the performance of this structure is close to the performance of the traditional bed of nails structure. The goal of project J2 has been to increase the gap-height between the two structures making up the glide-symmetric holey EBG structure. This increase in height has been achieved by rotating the unit cell of the glide symmetric holey EBG structure by 45 degrees and using multiple structures, each covering a span in the stopband.

This technology can be further improved by investigating how better transitions between the filter and the connected waveguide can be done. This will lead to the overall filter being more compact, which then means less mass needs to be placed in a satellite-rocket. Furthermore the filter can be even more compact by minimizing the gap height between the two plates making up the unit cell structure. Minimizing the gap height will result in a filter with better filtering properties as well as a more compact filter format.

IMPACT ON SOCIETY AND ENVIRONMENT

Novel antenna technology has a profound impact on both society and the environment. Antenna technology is a cornerstone in an interconnected world, as it makes quick communication possible, even in remote locations. Satellites with antennas broaden the coverage of a connected system, such as the GPS satellites, and could even enable connections to the internet from space. Low Earth Orbit (LEO) satellites give the opportunity for a stable internet connection for people in locations where the telecom network is not as developed. Undeveloped countries will not have to build up whole infrastructures to accommodate the inhabitants' need for the internet since there will be satellites orbiting the whole planet. An internet connection can be seen as a part of a human's right to free speech and satellites could make the world more equal.

Antenna technology also makes widespread usage of IoT possible, as 5G/6G allows for more bandwidth for sending data from IoT sensors to central computing hubs. This allows IoT devices to use less power. IoT brings both societal and environmental benefits. Security solutions would allow for constant monitoring and quick contact with emergency services. With smart

healthcare devices remote healthcare could become common for people who are in constant need of care. Smart houses could be saving energy by being temperature regulated from a distance.

The environmental effect caused by antenna technologies mostly comes from its energy consumption. In the new era of 5G where antennas have a shorter range and higher losses, more antennas need to be placed at shorter intervals from one another. It is also a physical rule that the attenuation of waves increases with higher frequencies. This increase in the number of antennas and higher frequencies will result in a higher total energy consumption. Since most countries around the world still rely on fossil fuels for their energy consumption the antennas will leave an environmental footprint by using that energy.

Antennas can be produced in such a way that they become environmentally sustainable and have a long lifespan. Many antennas however are connected to a product or system that have a much shorter lifespan than the antenna. Every wirelessly connected device has antennas, and when the device is discarded so is the antenna. For the economy, this is advantageous while from an environmental point of view products should be used during their entire lifetime. Similarly, LEO satellites need to be replaced after a couple of years due to deviation from orbit while the antennas onboard are still functioning.

Antennas are used for surveillance, for example, to track mobile phones or wearables. Another common use of antennas is in radar systems used by the military. Meteorological satellites can be used for tracking weather patterns and climate changes as well as to detect forest fires and dust storms. Military satellites can be used to monitor military or civilian activity. There are examples of where surveillance can become a human rights issue. The right to free speech can become threatened when governments choose to monitor civilian activity with the motive of shutting down opposing groups or individuals. On the other hand, surveillance of civilians can be used to prevent acts of terror and other threats to the population. Despite the problems mentioned here, we think that novel antenna technology will make the world a better place under the correct regulations.

3D-Printed Geodesic Luneburg Lens Antenna With Novel Patch Antenna Feeding

Elin Berglund and Sandis Freimanis

***Abstract*—With the roll out of new technologies and the world becoming more connected, there is a rising demand for higher bandwidth and new frequency bands. To meet the demand, higher frequencies are used in new communication systems. Higher frequencies come with the need for new antenna designs and one promising type of antenna is the lens antenna. In this paper, a modulated geodesic Luneburg lens with a novel feeding method is proposed for use between 8-10 GHz. Furthermore, the manufacturing of the lens explores the possibility of 3D printing as a method of producing cheap antennas.**

The paper verifies the viability of using a patch antenna and horn as a feeding method for a parallel-plate waveguide lens. First the lens is modeled and simulated in *CST Microwave Studio* and is then 3D-printed in PLA plastic and taped with copper tape. The antenna achieves -5 dB S_{11}-parameter between 8-10 GHz. The antenna also achieves $\pm 60^\circ$ scanning in the azimuth plane. The antenna achieves a HPBW of 15°.

***Sammanfattning*—Med utvecklingen av nya tekniker och en värld som blir allt mer digital är efterfrågan på större bandbredd och nya frekvensband hög. För att möta efterfrågan används högre frekvenser i nya kommunikationssystem. Med användningen av högre frekvenser behövs nya antenndesigner och en lovande typ av antenn är linsantennen. I den här artikeln föreslås en modulerad geodesic Luneburg lins med en ny typ av matningsmetod för användning mellan 8-10 GHz. För tillverkningen av linsen utforskas 3D-printning som en billig och enkel metod.**

Artikeln verifierar användningen av en patch-antenn och ett horn som matningsmetod för en lins av parallella metallplattor. Först simuleras linsen i *CST Microwave Studio* och 3D-printas sedan i PLA-plast och tejpas med koppartejp. Antennen åstakommer -5 dB i S_{11}-parameter mellan 8-10 GHz. Antennen har en skanning av $\pm 60^\circ$ i azimut-planet och har en HPBW av 15°.

***Index Terms*—Luneburg lens, patch antenna, geodesic, transformation optics, 3D-printing**

Supervisors: Pilar Castillo Tapia, Sarah Clendinning, Oscar Quevedo Teruel

TRITA number: TRITA-EECS-EX-2021:169

I. Introduction

With modern communication increasingly requiring more data bandwidth, the demand for new frequency bands is high. As 5G and eventually, 6G is rolled out with higher frequencies and shorter wavelengths this causes a problem with attenuation. Signals with short wavelengths lose strength faster when traveling over longer distances, therefore higher directivity will be needed for the antennas. Simultaneously, antennas operating in those frequencies can be made smaller, making them cheaper and giving opportunities for new designs. A solution to increase the directivity of the antenna is to use a lens. At low frequencies, lenses are big and expensive to manufacture but decrease in size proportional to the increase in frequency. Lens antennas can focus and disperse electromagnetic waves in the same way as optical lenses. In that way, they can be combined with other antennas to increase directivity [1].

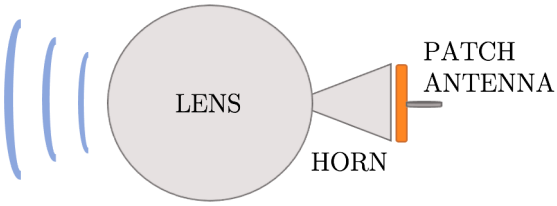


Fig. 1. Sketch of the full system to be designed. It is fed from the patch antenna into a horn, which then leads the waves to the lens. Plane waves radiate from the lens opening.

One type of lens antenna is the Luneburg lens [2]. It was proposed by R.K. Luneburg in 1944 and has recently gained attention by [3], [4], [5] and [6] for use in microwave applications. A Luneburg lens has the property of focusing a plane wave on the opposite surface of the lens [2]. The advantage of a Luneburg lens antenna is that azimuthal scanning can be achieved without scan losses. Different alternatives to the dielectric spherical Luneburg lens have been studied. In [3], a fully metallic and flat Luneburg lens was proposed with the use of glide-symmetric holes loaded in a parallel plate. Glide-symmetric structures are periodic structures that do not change with mirroring or transformation. The holes were used to give the lens the refractive index properties of the Luneburg lens. Having a fully metallic lens reduces losses that come with dielectrics and are easier to design and manufacture. In [4] it was shown that the refractive index of a dielectric lens could be mapped onto a geodesic curve, making it possible to construct a curved flat lens with the same properties as a dielectric lens. A 2-D Luneburg lens may be used for focusing beams in 1-D as in [6], where a fully metallic geodesic Luneburg lens antenna was proposed. The lens was constructed with a parallel plate waveguide (PPW). To reduce the height of the lens it was folded at two points while keeping the focusing properties. Height reduction by folding is further explored in [5] where another method is introduced using ray tracing to optimize the geodesic lens. A similar design was implemented in [7] where a half geodesic PPW Luneburg lens was made. In this design, the lens can be made significantly smaller with the cost of

a reduced angular scanning range. In [8], a PPW Luneburg lens is presented based on a substrate integrated holes (SIH) metasurface with square holes. The SIH holes connect the substrate to the ground plane. Typically in literature the lenses are fed through a waveguide using coaxial feeding [3], [5]- [8]. In this paper, a novel feeding method using a microstrip patch antenna is presented.

In this paper a modulated geodesic Luneburg lens is designed for the X-band (8-10 GHz). The lens is fed from a patch antenna through a horn, combining three design elements. A sketch of the system is presented in Fig. 1.

II. Theory

A. Geodesic Luneburg Lens

The gradient refractive index $n(r)$ of a Luneburg lens is dependent on the normalized radius r given by the equation:

$$n(r) = \begin{cases} \sqrt{2-r^2}, & 0 \leq r \leq 1 \\ 1, & 1 > r \end{cases} . \tag{1}$$

In [5] and [6] the geodesic Luneburg lens is described in differential form as:

$$\frac{dz}{d\rho} = -\sqrt{\left(\frac{1}{2} + \frac{1}{2\sqrt{1-\rho^2}}\right)^2 - 1} \tag{2}$$

with the height z and radius ρ. The equation was solved numerically in MATLAB by rewriting it as:

$$z(i) = z(i-1) - \left(\sqrt{\left(\frac{1}{2} + \frac{1}{2\sqrt{1-\rho(i)^2}}\right)^2 - 1}\right) d\rho(i) \tag{3}$$

where $d\rho(i) = \rho(i) - \rho(i-1)$. The equation is solved iteratively for z by setting $z(1) = \frac{1+\sqrt{5}}{2} - 1 + h_1$, where the constant $h_1 = 0.0145$ is added so that $z(end) = 0$.

In [4] a method to map lenses onto curved surfaces is proposed. The method maps a lens with a refractive index $n_1(r)$ to a curved lens with refractive index $n_2(\theta)$. The geometries are shown in Fig. 2. To verify that (3) in fact calculates a geodesic Luneburg lens with a homogeneous refractive index, the method in [4] is applied. For the geodesic lens to have the same focusing properties as the planar Luneburg lens with refractive index $n_1(r)$, the ray paths s_1, s_2 and l_1, l_2 shown in Fig. 2 are set to be equal. In [4] the equal ray paths s_1 and s_2 are given by:

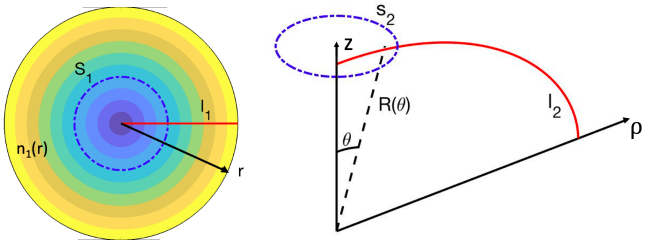
$$\begin{aligned} s_1 &= s_2 \\ n_1(r) 2\pi r &= n_2(\theta) 2\pi R(\theta) sin(\theta) \end{aligned} \tag{4}$$

furthermore the ray paths l_1 and l_2 are given by:

$$\begin{aligned} l_1 &= l_2 \\ n_1(r) dr &= n_2(\theta) \sqrt{R(\theta)^2 + R'(\theta)^2} d\theta \end{aligned} \tag{5}$$

The refractive index $n_1(r)$ is given by (1). From Fig. 2b we have that $R(\theta) = \sqrt{z^2 + \rho^2}$. By equating and rewriting (5) and (4) in discrete form we get:

$$r(i) = r(i-1)\left[1 - \frac{\sqrt{R(\theta(i))^2 + R'(\theta(i))^2} d\theta}{R(\theta(i)) sin(\theta(i))}\right]^{-1} \tag{6}$$



(a) Flat Luneburg lens (b) Geodeseic Luneburg lens

Fig. 2. Ray paths on flat Luneburg lens and geodesic Luneburg lens

where $d\theta = \theta(i) - \theta(i-1)$ and where $R'(\theta(i))$ is the the derivative of $R(\theta(i))$ and is defined as:

$$R'(\theta(i)) = \frac{R(\theta(i)) - R(\theta(i-1))}{\theta(i) - \theta(i-1)} \tag{7}$$

Further we have that:

$$n_2(i) = \frac{n_1(i) * (r(i) - r(i-1))}{\sqrt{R(\theta(i))^2 + R'(\theta(i))} d\theta} \tag{8}$$

which when solved numerically in MATLAB with 400 data points gives a $max(n_2(\theta)) = 1.0035$. As the amount of data points is increased, deviation of $n_2(\theta)$ is reduced. The choice of 400 data points is further discussed in Section VII.

B. Wave Propagation

Transversal electric and magnetic (TEM) mode wave propagation is characterized by having no electric and magnetic fields in the direction of propagation. Transversal electric (TE) and transversal magnetic (TM) are characterized by having non-zero magnetic and electric field in the propagating direction respectively. A TEM wave can not exist in a single conductor waveguide, but can exist in a PPW or coaxial cable. A time-harmonic wave propagating along the z-axis can be written as:

$$\bar{E}(x,y,z) = [\hat{x}e_x(x) + \hat{y}e_y(y) + \hat{z}e_z(z)]e^{-j\beta z} \tag{9a}$$

$$\bar{H}(x,y,z) = [\hat{x}h_x(x) + \hat{y}h_y(y) + \hat{z}h_z(z)]e^{-j\beta z} \tag{9b}$$

where β is the propagation constant. Further the wave number is defined as:

$$k = \omega\sqrt{\mu\varepsilon} \tag{10}$$

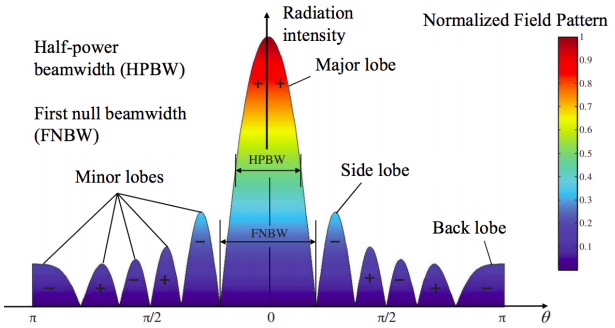


Fig. 3. Example of a radiation pattern Source: [9]

where ω is the angular frequency, μ is the permeability and ε is the permittivity.

For a rectangular waveguide the propagation constant β is equal to:

$$\beta = \sqrt{k^2 - k_c^2} \tag{11}$$

where k_c is the cut-off wave number, which for TE and TM modes in a rectangular waveguide is defined as:

$$k_c = \sqrt{\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2} \tag{12}$$

where a is the width of the waveguide and b is the height. Furthermore m and n are any non-negative integer. In order for a wave to propagate, β must be real. This is true when:

$$k > k_c = \sqrt{\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2} \tag{13}$$

and leads to each combination of m and n having a cut-off frequency $f_{c_{mn}}$, which is given by:

$$f_{c_{mn}} = \frac{k_c}{2\pi\sqrt{\mu\varepsilon}} = \frac{1}{2\pi}\sqrt{\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2}. \tag{14}$$

If $a > b$ then the lowest order mode is TE_{10} and has a cutoff frequency:

$$f_{c_{10}} = \frac{1}{2a\sqrt{\mu\varepsilon}} \tag{15}$$

where $m = 1$ and $n = 0$. The lowest order TM mode is TM_{11}, and has cutoff frequency:

$$f_{c_{11}} = \frac{1}{2\pi\sqrt{\mu\varepsilon}}\sqrt{\left(\frac{\pi}{a}\right)^2 + \left(\frac{\pi}{b}\right)^2} \tag{16}$$

which is a higher frequency than the cutoff frequency for the TE_{10} mode [10].

C. Antenna Radiation Pattern

To be able to quantify an antenna's performance, some antenna characteristics are explained. An example of an antenna radiation pattern is shown in Fig. 3. The radiation pattern shows how the radiation intensity from an antenna is directed in space. The largest lobe is called "main lobe" while the the smaller lobes on the sides are called "side lobes". The lobe directly behind the main lobe is called "back lobe". The angle where the radiation intensity is half, is called the "half-power bandwidth" (HPBW). Directivity is a measure of the ratio of radiation intensity in a direction, divided by the average radiation intensity in all directions. This can be written as:

$$D = \frac{4\pi U}{P_{rad}} \tag{17}$$

where D is the antenna's directivity, U the radiation intensity and P_{rad} is the average radiation intensity in all directions. While directivity only describes the relative strength of radiation intensity in all directions, *gain* is also a measure of the efficiency of the antenna. Gain is defined as:

$$G = 4\pi\frac{U(\theta,\phi)}{P_{in}} \quad (dimensionless) \tag{18}$$

where G is gain, $U(\theta,\phi)$ is the radiation intensity in a given direction and P_{in} is the input power. Further if losses from reflections from the transmission line is taken into account *realized gain* is be defined. Realized gain, G_{re}, is defined as:

$$G_{re}(\theta,\phi) = e_r G(\theta,\phi) \tag{19}$$

where e_r is the mismatch efficiency of the antenna [9].

D. Scattering Parameters

Scattering parameters or S-parameters are elements in the scattering matrix for a system:

$$\begin{bmatrix} V_1^{ref} \\ V_2^{ref} \\ . \\ . \\ . \\ V_N^{ref} \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} & . & . & . & S_{1N} \\ S_{21} & & & & & \\ . & & & & & \\ . & & & & & \\ . & & & & & \\ S_{N1} & S_{N2} & . & . & . & S_{NN} \end{bmatrix} \begin{bmatrix} V_1^{inc} \\ V_2^{inc} \\ . \\ . \\ . \\ V_N^{inc} \end{bmatrix}. \tag{20}$$

The scattering matrix describes the system in incident and reflected waves as seen from its N ports. The elements in the matrix are defined as:

$$S_{ij} = \frac{V_i^{ref}}{V_j^{inc}} \tag{21}$$

where V_j^{inc} is the amplitude of the incident voltage at port j and V_i^{ref} is the reflected voltage amplitude from port i. For example if port 1 is the input port then, S_{11} defines the voltage reflection coefficient ($S_{11} = \frac{V_1^{ref}}{V_1^{inc}} = \Gamma_{in}$) and S_{21} the voltage gain at port 2. S_{11} is in simple terms the reflection from port 1 back to itself and S_{21} is the radiation transmitted to port 2 from port 1. S-parameters are calculated in dB and can be measured using a Vector Network Analyzer (VNA) [11].

E. Microstrip Antennas

Microstrip antennas, also called patch antennas, are small, cheap, and easy to manufacture. A conventional rectangular patch antenna consists of a thin ground plane under a thin metallic patch. Between the ground plane and the patch is a substrate consisting of a dielectric material, normally with a dielectric constant (ε_r) between 2.2 and 12. The properties of the substrate and the dimensions of the patch decide the properties of the antenna. Fig. 4 shows a simple patch antenna with radiating fields. The feeding can be done in different

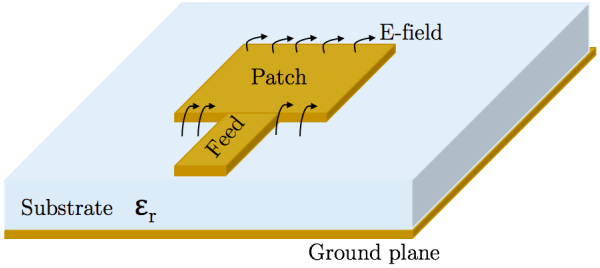


Fig. 4. Simple patch antenna

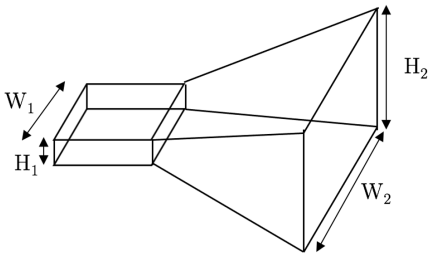
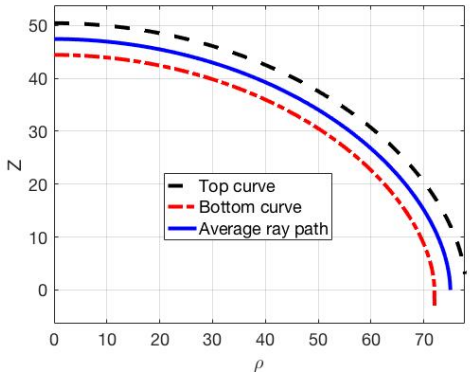


Fig. 5. Pyramidal horn

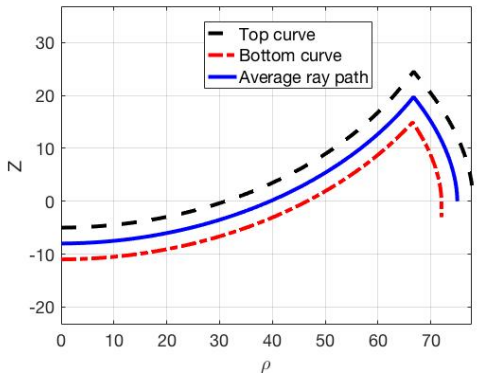
ways, the most common are coaxial feeding through the substrate and microstrip line feeding which is done in Fig. 4. Patch antennas are known for narrow bandwidth and low efficiency but can be improved with different design methods, for example by introducing slots in the patch or increasing the substrate height. Patch antennas are usually manufactured by photo-etching the radiating components on the substrate [12].

F. Horn Antennas

Horn antennas have been in use since the 1800s and are widely used as a feed element for their easy construction and high gain [13]. There are several types of horn antennas. One of the most common types are pyramidal horns. These are flared in both the E-plane and the H-plane, and are fed by a rectangular waveguide [13]. Fig. 5 shows a sketch of a pyramidal horn antenna.



(a) Geodesic Luneburg curve



(b) Modulated lofted curve

Fig. 6. Lens curves with average ray path set to 2.5 wavelength radius at 10 GHz.

III. Lens design

The lens is designed in four steps. First the design and calculations of the geodesic Luneburg lens are made. Secondly the design of the flare is made. Thirdly the lens is modulated to reduce the height and at last the chamfers for the lens are designed.

A. Geodesic Luneburg lens

The geodesic Luneburg lens curve is calculated from (3). The curve is the offset above and under the average ray path as seen in Fig. 6a. The length of the offset is discussed in Section III-B.

B. Lens Cavity and Waveguide

The height of the lens cavity is set to be smaller than a quarter-wavelength of the highest frequency, so that only the fundamental TEM mode can propagate [5]. As the maximum frequency for the system is aimed for 12 GHz, the lens cavity height $h_L = 6$ mm. The size of the waveguide is designed so that only the fundamental TE_{10} mode is excited. The height of the waveguide from the horn is set to the same height as the lens cavity. The width of the waveguide is set to be smaller than the wavelength of the highest frequency, 12 GHz. The waveguide width is set to 24 mm.

C. Design of flare

In order to reduce reflections at the receiving end of the antenna, a flared opening is introduced. The flare chosen is similar to the flare used in [6]. Instead of using a sinusoidal flare as in [6], a straight flare is used for ease of manufacturing, and instead of having the flare cover 220° of the lens, the flare covers 200° to make room for mounting the top and bottom plate together. The height of the flare is optimized to be small but still achieving good scanning in the elevation plane. The height of the flare is set to 20 mm. The length of the flare was in turn restricted by the 3D-printer used. The 3D-printer could print objects up to 300x300 mm. With a lens diameter of 154 mm, this restricted the size of the flare to 73 mm. The

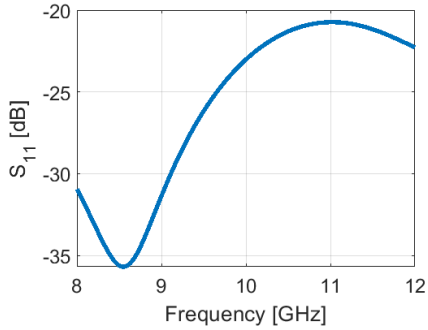


Fig. 7. S_{11}-parameter for simulated flare

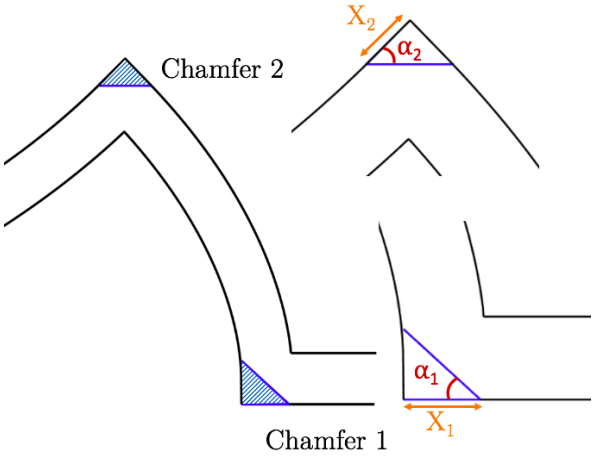
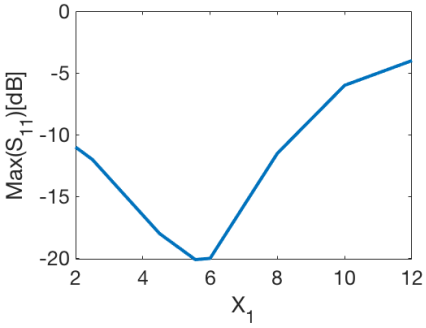


Fig. 8. Parameters of chamfer 1 and 2

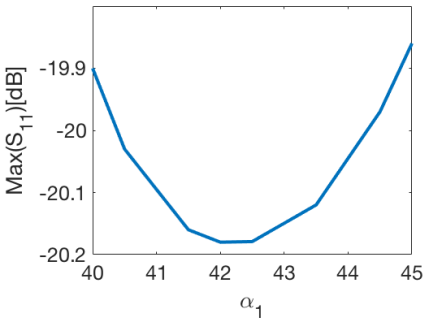
optimal length for the horn of the flare is found by setting the height to 20 mm, and then a parameter sweep is performed for different lengths. The optimal length is found to be 38 mm. The S_{11}-parameters are seen in Fig. 7. The S_{11}-parameters are under -20 dB for 8-12 GHz.

D. Modulation of lens

In [6] the height of the lens is reduced by folding the lens at two points. Due to the relationship between the ρ and z being kept, this does not change the focusing properties of the lens. In this project, the lens is folded at one point with the goal of minimizing the total height of the lens. For just reducing the height of the lens the optimal method would be to fold the lens in half, reducing the height by 50%. However, the height of the flare in section III-C is taken into consideration. The flare is symmetric in the z direction with center at zero, as shown in Fig. 6a. Folding the lens in half would therefore result in extra added height from the flare. In section III-C the flare is set to 20 mm, meaning it adds 10 mm of height in the negative z direction. To not have any extra height added by the flare, the curve is folded below zero. In Fig. 6b the curves in Fig. 6a are folded at z = 24.5 mm, making the lowest point of the lens at z = -11 mm and the highest point at z = 24.5 mm. This is a height reduction of 34 %.

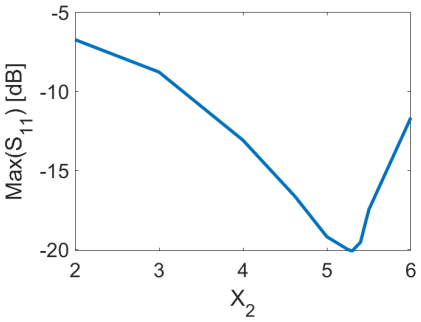


(a) Sweep for X_1 with $\alpha_1 = 45°$

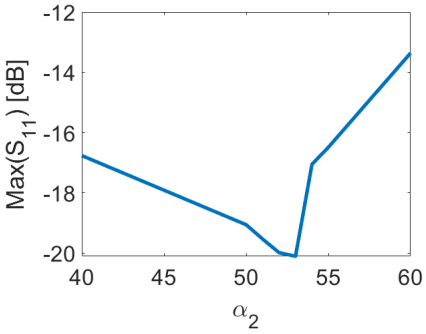


(b) Sweep for α_1 with $X_1 = 5.56$ mm

Fig. 9. Parameter sweep results for chamfer 1



(a) Sweep for X_2 with $\alpha_2 = 52°$

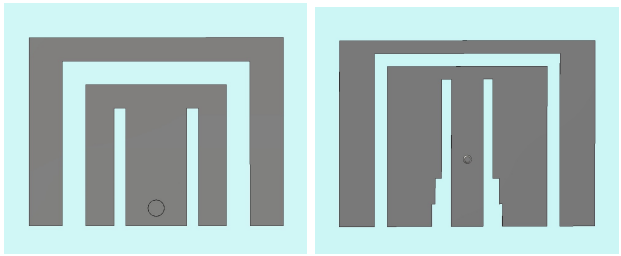


(b) Sweep for α_2 with $X_2 = 5.3$ mm

Fig. 10. Parameter sweep results for chamfer 2

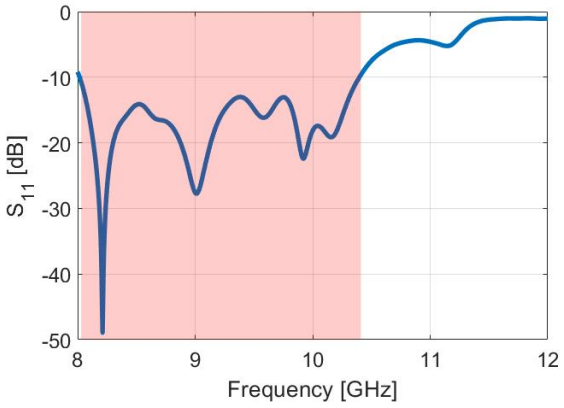
E. Chamfer design

To reduce reflections, chamfers are added to the lens at two places, see Fig. 8. The first chamfer is located at the edge of the lens, where it meets the waveguide and horn on one side

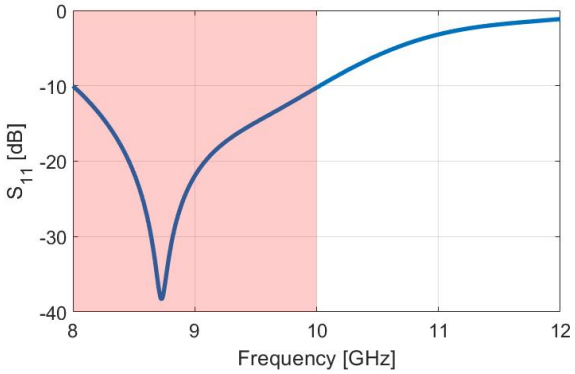


(a) E-shaped patch (b) E-shaped patch with extra slots

Fig. 11. E-shaped patch antennas



(a) E-patch



(b) Stacked patch

Fig. 12. S_{11}-parameters for patch antennas

and the flare on the other. The second chamfer is located at the top of the lens, where the lens is folded. To decide the optimal values for the parameters in Fig. 8, simulations are made in *CST Microwave Studio*. For chamfer 1 simulations are made with the non modulated lens. First different values of X_1 with $\alpha_1 = 45°$ is simulated. In Fig. 9a the maximum of the S_{11} parameters for different values of X_1 have been plotted. From the results it is seen that $X_1 = 5.56$ mm is the

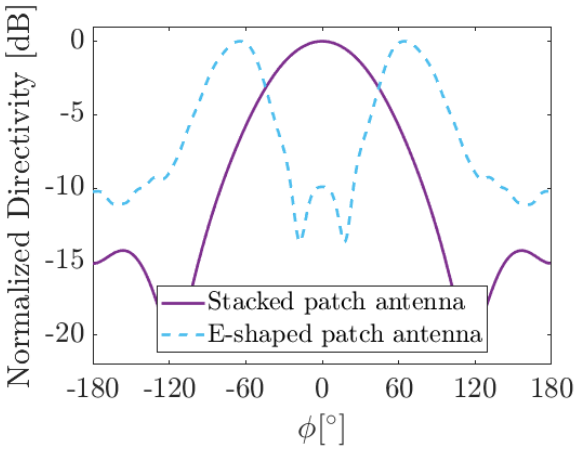


Fig. 13. Normalized farfields of E-shaped and stacked patch antennas

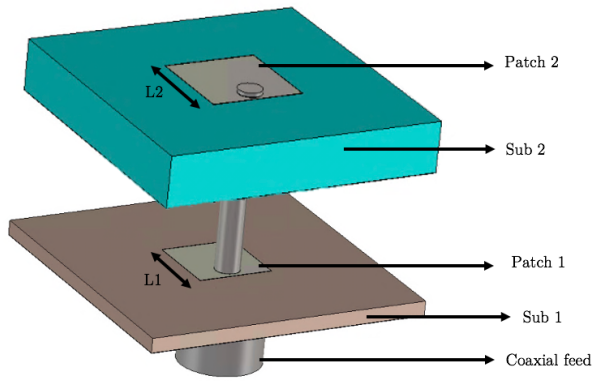


Fig. 14. Sketch of stacked patch antenna

optimal value with $\alpha_1 = 45°$. Further the same thing is done for α with $X_1 = 5.56$ mm. The results are plotted in Fig. 9b. The optimal value for α when $X_1 = 5.56$ mm is seen to be $42.5°$.

For chamfer 2, simulations are made with the modulated lens and chamfer 1, with the optimal values simulated above. First different values for X_2 is simulated with $\alpha_2 = 52°$. The results are shown in Fig. 10a and the optimal value is seen to bee $X_2 = 5.3$. Then α_2 is simulated with $X_2 = 5.3$, the results are shown in Fig. 10b. The optimal value of α_2 with $X_2 = 5.3$ is seen to be $53°$.

IV. Design of Patch Antenna

The materials available for the patch antenna are a substrate RT6002 with thickness of 1.52 mm and an dielectric constant, ε_r, of 2.94 [14], and a foam with ε_r close to 1. The coaxial connector used is RND 205-00498 [15]. The goal of the patch antenna is to achieve a bandwidth of 40 % between 8 - 12 GHz. In [16] a bandwidth between 8.34 and 13.86 GHz is achieved using an E-shaped patch and a substrate with $\varepsilon_r = 1.03$ and height 3.2 mm. The antenna is fed using a coaxial feeding. A patch antenna with the same E-shaped patch is designed in CST (see Fig. 11a) and simulated for different parameters, trying to achieve a high bandwidth with the available material. Four extra slots are added to the patch with the goal of increasing the bandwidth, see Fig. 11b. The results for this antenna show the S_{11}-parameters in Fig. 12a with a bandwidth between 8 - 10.4 GHz and the radiation pattern in Fig. 13. As the patch antenna will be connected to a horn, as described in

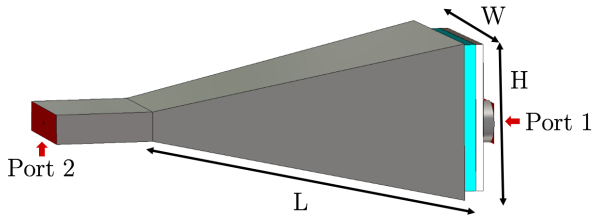


Fig. 15. Model of horn and patch antenna in CST

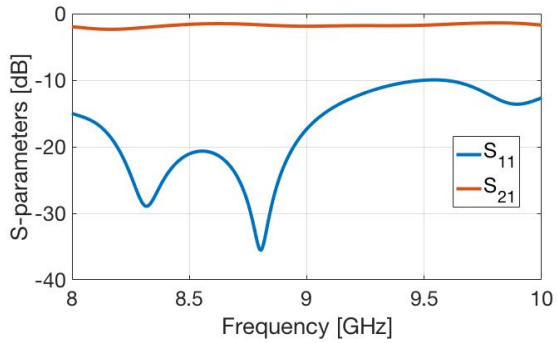


Fig. 16. S-parameters for horn and patch antenna

section V, a single main lobe directed straight into the horn is preferred, making the E-patch a deficient solution.

Another method for achieving a larger bandwidth is to stack two patch antennas on top of each other. The top patch will work as a parasitic patch to the lower driven patch and will, if matched properly, enhance the bandwidth [17]. To achieve this, two square patch antennas are created and stacked on top of each other, see Fig. 14. The first layer consists of the ground plane, a substrate, Sub 1 with $\varepsilon_r = 2.94$ and height 1.52 mm, and a patch, Patch 1 with length L1. The second layer consists of a foam, Sub 2, with ε_r close to 1 and height 5.5 mm, and a patch, Patch 2 with length L2. The foam is picked for the second layer to increase the bandwidth. The bandwidth is proportional to the height of the substrate as well as inversely proportional to $\sqrt{\varepsilon_r}$ [12]. A coaxial feeding is connected to the lower patch and the feeding probe runs through both layers. There is no gap between the layers.

L1 and L2 are found by simulating them for different values. It is found that L1 had a greater impact on the bandwidth than L2. A good match in resonance frequency was found with L1 = 9.4 mm and L2 = 10.7 mm. With these values the S_{11}-parameters presented in Fig. 12b shows a 20% bandwidth between 8-10 GHz. The far field pattern of the antenna is presented in Fig. 13.

The stacked patch is picked for the system due to its single main lobe in the radiation pattern. The bandwidth between 8-10 GHz is deemed to be acceptable for the system, even if it does not meet the preliminary goal. The design and simulations are here on made with the knowledge that the system will operate between 8-10 GHz.

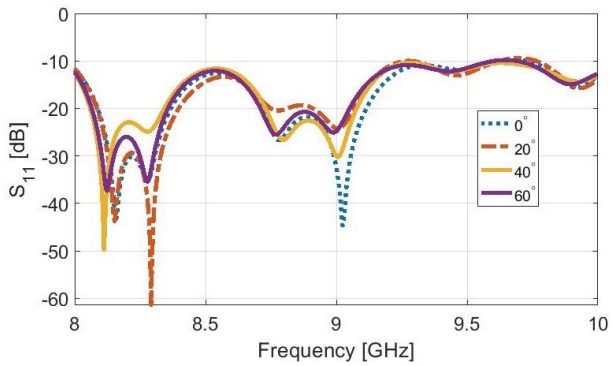


Fig. 17. S_{11}-parameters for simulation of system for different scanning angles

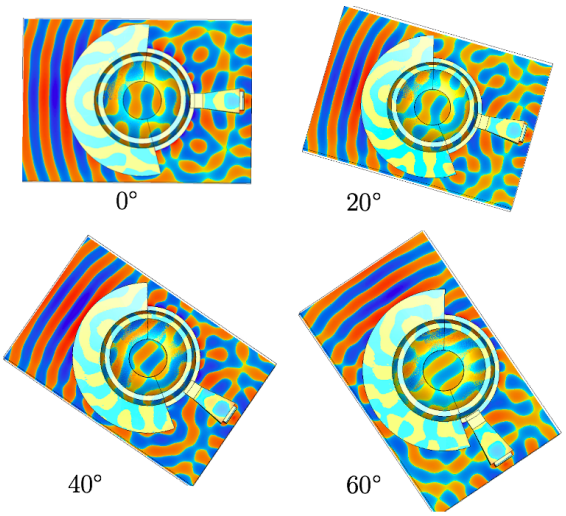


Fig. 18. Simulated E-field for different scanning angles

V. Design of Feeding Horn

To efficiently transfer the signal from the patch antenna to the waveguide, a horn antenna is used. Horn antennas are waveguides which are tapered at the end. Depending on which sides are tapered, horn with different characteristics can be produced [13]. Since the patch antenna designed in section IV is both wider and higher than the waveguide used, a pyramidal horn antenna was chosen. A pyramidal horn has all of its sides tapered. As a reference, a method for designing pyramidal horn antennas from [13] was used.

Fig. 15 shows the model of the horn and the patch antenna made in CST. Because the horn and patch antennas are placed

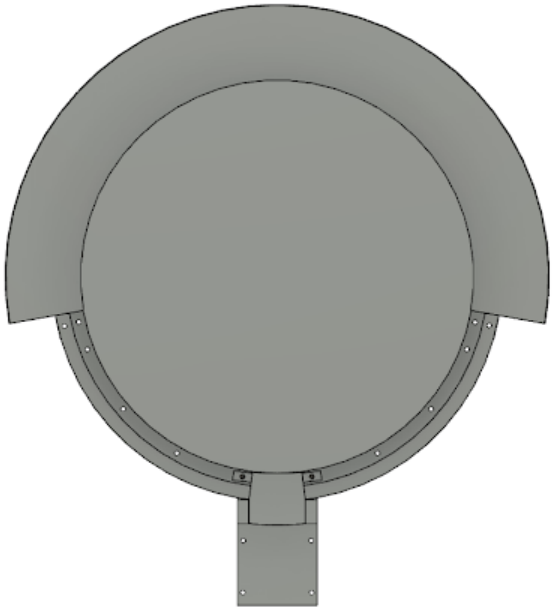


Fig. 19. Top view of lens and sliding mechanism in *Autodesk Fusion 360*

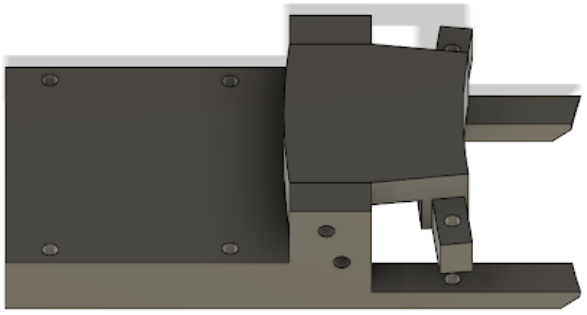


Fig. 20. Side view of sliding mechanism and waveguide in *Autodesk Fusion 360*

close to each other, the interaction changed the characteristics of the horn antenna. To optimize the horn, parameter sweeps were used to optimize the height, length and width. The optimized parameters were $H = 40$ mm, $W = 32mm$ and $L = 60mm$. The S_{21}- and S_{11}-parameters for the patch and horn combination can be seen in Fig. 16.

VI. Simulation of system

The antenna is simulated in *CST Microwave Studio* using the *Time-domain solver*. Simulations are made of the whole system for different scanning angles. Fig. 17 shows the S_{11}-parameters. The simulation shows that the system almost achieves -10 dB S_{11}-parameters for 8-10 GHz. This is deemed as satisfactory results to manufacture the whole system. Fig. 18 shows the E-field for different scanning angles. It is clearly visible that the antenna radiates a plane wave on the opposite side of the waveguide. Since symmetry is assumed, this is presumed to be true for all angles ($-60° \rightarrow 60°$).

VII. Manufacturing

A. CAD model

The lens and horn is 3D-modeled with Autodesk Fusion 360. For the geodesic curve 400 data points are used from solving (3). The amount of data points is chosen as a compromise between accuracy of the lens curve and software usability. A top view of the lens is shown in Fig. 19. A sliding mechanism is designed so that the horn can be placed at multiple positions

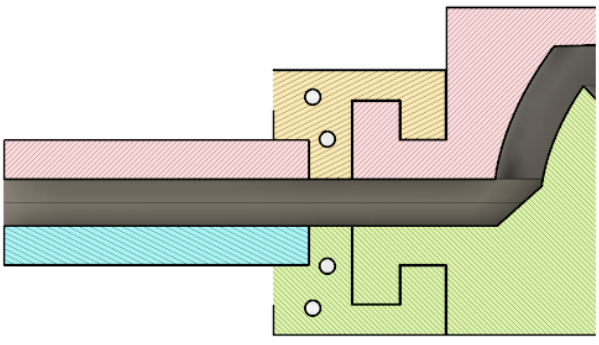
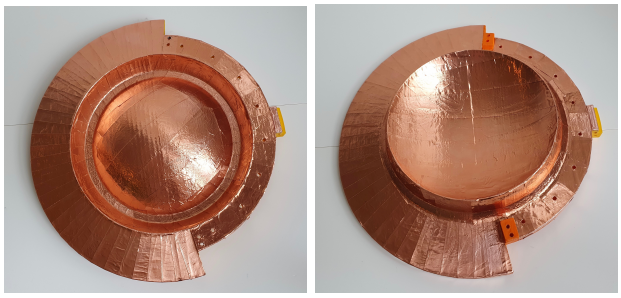


Fig. 21. Cross section of lens, waveguide and sliding mechanism in *Autodesk Fusion 360*



(a) Top plate of lens (b) Bottom plate of lens

Fig. 22. Lens taped with copper tape

relative to the center of the flare. The horn is possible to be fastened with screws at $0°$, $\pm 20°$, $\pm 40°$ and $\pm 60°$. The sliding mechanism in Fig. 20 consists of a groove and a component which can slide in the groove and is joined together with the waveguide. The cross section of the sliding mechanism and waveguide is seen in Fig. 21.

B. 3D-printing

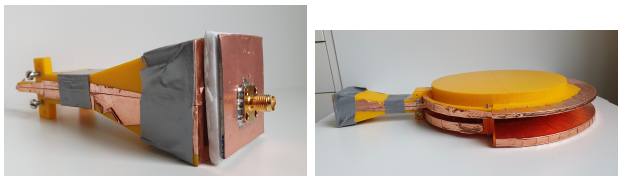
The lens is 3D printed in polyaktid PLA and then taped with copper tape to get a metallic surface. Fig. 22 shows the taped lens. Fig. 23a shows the horn and the patch antenna. Fig. 23b shows the horn and the lens screwed together. Fig. 24 shows a cut view of the printed lens.

C. Patch Antenna

To make the patch antenna a printed circuit board (PCB) with the substrate RT6002 and dimensions 40 x 40 mm is used. It is decided that the patch antenna should cover the opening of the horn antenna to reduce leakage and for easier placement. The copper is removed from one side of the PCB and is kept on the other side for the ground plane. Two pieces of copper tape are used as the patches with dimensions from section IV. The patches are fixed on either side of a piece of foam, with a height of 5.5 mm and the same size as the PCB. The foam is then placed on top of the PCB, at the copper-free side, and a hole is drilled for the feeding probe. The coaxial connector with the feeding probe is then soldered to the PCB and to the top patch. Fig. 25 shows the finished patch antenna.

VIII. Measurement results

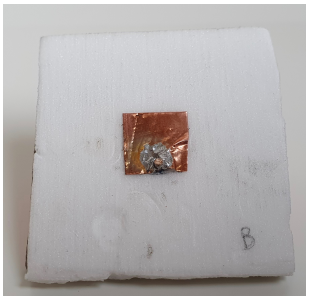
To measure the S-parameters of the antenna a VNA was used. The set up for the measurement is shown in Fig. 26. The S_{11}-parameters was measured for the patch antenna alone



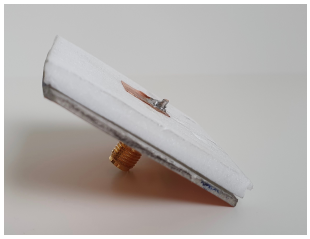
(a) Horn and patch antenna (b) Lens and horn

Fig. 23. Horn and patch joined together and full system assembled.

Fig. 24. Cut view of 3D printed lens



(a) Top view of patch antenna



(b) Side view of patch antenna

Fig. 25. Finished patch antenna

and for different scanning angles for the whole system. The results are presented in Fig. 27 and 28 respectively. In Fig. 17 the simulated S_{11} parameters for different scanning angles can be seen. Fig. 29 shows the S_{11}-parameters for the simulated system and measured system at a 0° scanning angle.

To measure the farfields the antenna was measured in an anechoic chamber. The setup is shown in Fig. 30. In the anechoic chamber the walls are covered in foam made of carbon fibers that absorb radio waves. In that way there are no reflections in the chamber, making an ideal environment for measurements. The result for the farfields are presented in Fig. 31 for 10 GHz. The realized gain for the system is seen in Fig. 32. It is seen that the measured radiation pattern is closer to the simulation at 10 GHz. The HPBW for α_f for the 10 GHz simulation is approximately 15°.

IX. Possible Improvements

To improve the antenna further for future work a discussion of possible improvements is made here.

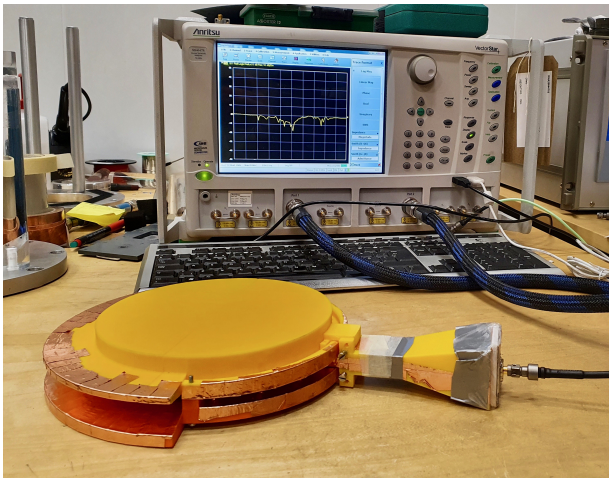


Fig. 26. Measuring set up for VNA

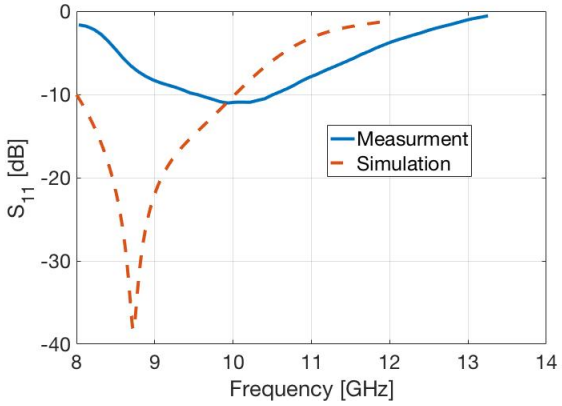


Fig. 27. Measured and simulated S_{11}-parameters for patch antenna

A. Order of design

The goal of this project was to design the antenna for use between 8-12 GHz. The design of the lens was therefore made with respect to 8 GHz as the lowest frequency and 12 GHz as the highest. When the patch antenna was to be designed, a bandwidth between 8-10 GHz was achieved. This meant that the lens was designed for a wider bandwidth than what the feeding could give. The size of the waveguide and the height of the lens cavity were designed after the highest frequency, 12 GHz, but could have been made differently for 10 GHz. If the project was to be remade, the feeding part could be designed first to avoid this.

B. Patch antenna

In the making of the patch antenna, copper tape was used to create the patches. The glue on the tape did not adhere well to the foam (Fig. 25) and therefore loosened when the drilling of the feeding hole was made and when the tape was soldered to the feeding probe. To prevent this the patches could be made with a more sturdy material or the tape could be drilled on its own and attached to the foam after the drilling.

When soldering the patch antenna only the top patch is soldered to the feeding probe. As seen in Fig. 27 the bandwidth

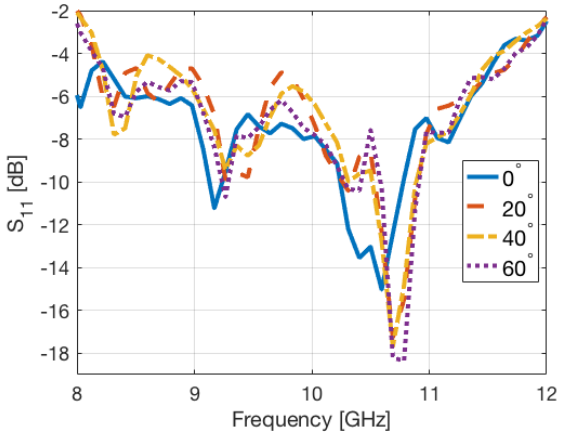


Fig. 28. S_{11}-parameters for system with different scanning angles

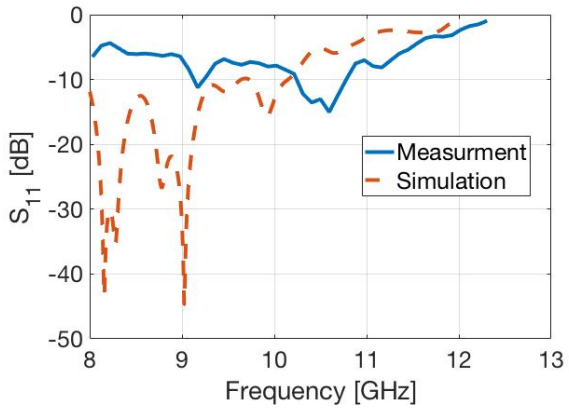


Fig. 29. Measured and simulated S_{11}-parameters for system at 0°

of the patch antenna was not as expected from the simulation results. Therefore, it is believed that because the lower patch was not soldered to the feeding probe, it did not radiate as expected. Hence an improvement would be to solder both patches to the feeding probe.

C. Sliding mechanism

The sliding mechanism for the waveguide did not work as designed. The waveguide did not slide well in the rails designed for the lens, see Fig. 19, 20 and 21. This was due to the plastic expanding when the lens was 3D-printed. To prevent this, margins could be added to the CAD model to count in for the expansion.

D. Horn and patch antenna

To reduce reflections and get better throughput from the patch antenna to the lens, it could be further explored how to best optimize the feeding horn and patch antenna. For example, the patch antenna's placement with respect to the horn could be explored, and a sinusoidal horn could be tested.

X. Conclusion

In this paper a 3D-printed modulated geodesic Luneburg lens with a novel feeding method for use in 8-10 GHz is designed and manufactured. A stacked patch antenna is used for the feeding through a pyramidal horn. A sliding mechanism is added for scanning in different directions. The 3D-printed antenna is covered with copper tape on the inside to achieve a metallic coating. The manufactured antenna achieved S_{11}-parameters of -5 dB over 8-10 GHz. The antenna is capable of azimuthal scanning in 7 positions between $\pm 60°$. The results seems to be limited by the patch antenna, that does not perform as expected. Therefore the model needs to be worked on further before real use. However the results show that the manufactured antenna is a good model for this feeding method.

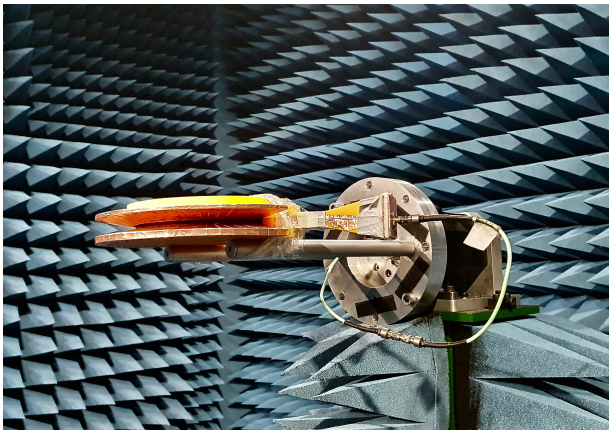


Fig. 30. Measurement set up in anechoic chamber

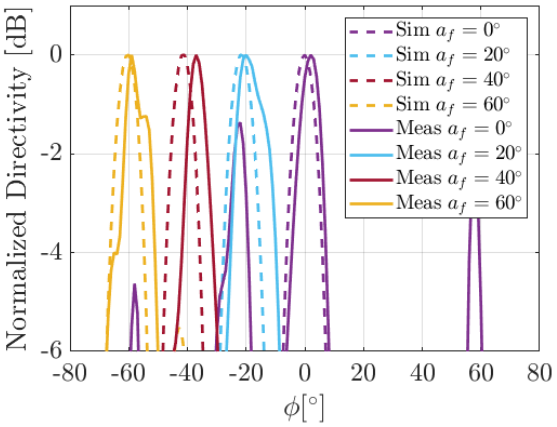


Fig. 31. Normalized radiation pattern for simulated and measured antenna at 10 GHz. α_f is the angle of the feed. Farfield cut $\theta = 90$.

XI. Future Work

The project shows that using a patch antenna in combination with a horn is a viable feeding method for a PPW geodesic lenses. For sub-terahertz systems, above 100 GHz, a coaxial cable feeding into a waveguide is difficult to implement. The lens designed in this paper is possible to be adapted for sub-terahertz as only the distance between the waveguides determines for which frequencies only the fundamental TEM mode is excited.

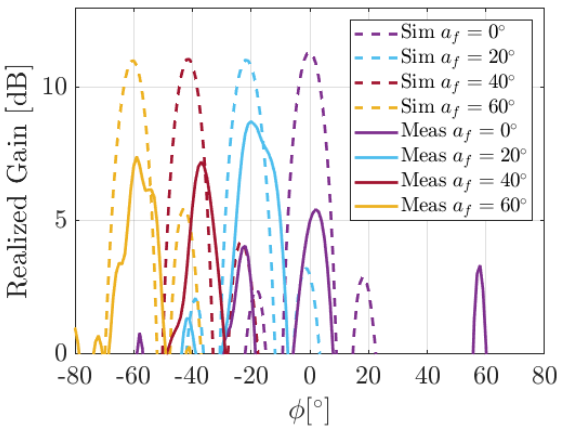


Fig. 32. Realized gain for the simulated and measured antenna at 10 GHz. α_f is the angle of the feed. Farfield cut $\theta = 90$.

ACKNOWLEDGMENT

The authors would like to thank the project's supervisors Pilar Castillo Tapia and Sarah Clendinning for valuable help and guidance. The authors would also like to thank Linus Backlund for use of his 3D-printer and help with using CAD software. At last the authors would like to thank the project members of J1b, for valuable discussions about the context and solutions to problems.

REFERENCES

[1] Y. Wang, J. Li, L. Huang, Y. Jing, A. Georgakopoulos, and P. Demestichas, "5G mobile: Spectrum broadening to higher-frequency bands to support high data rates," *IEEE Vehicular Technology Magazine*, vol. 9, no. 3, pp. 39–46, Sept 2014.

[2] R. Luneburg, *Mathematical theory of optics*. Brown University Press, Providence, RI, 1944, pp. 208–213.

[3] O. Quevedo-Teruel, J. Miao, M. Mattsson, A. Algaba-Brazalez, M. Johansson, and L. Manholm, "Glide-symmetric fully metallic Luneburg lens for 5G communications at Ka-band," *IEEE Antennas and Wireless Propagation Letters*, vol. 17, no. 9, pp. 1588–1592, Sept 2018.

[4] R. C. Mitchell-Thomas, O. Quevedo-Teruel, T. M. McManus, S. A. R. Horsley, and Y. Hao, "Lenses on curved surfaces," *Opt. Lett.*, vol. 39, no. 12, pp. 3551–3554, Jun 2014. [Online]. Available: http://ol.osa.org/abstract.cfm?URI=ol-39-12-3551

[5] N. Fonseca, Q. Liao, and O. Quevedo-Teruel, "Equivalent planar lens ray-tracing model to design modulated geodesic lenses using non-euclidean transformation optics," *IEEE Transactions on Antennas and Propagation*, vol. PP, pp. 1–1, Jan 2020.

[6] Q. Liao, N. J. G. Fonseca, and O. Quevedo-Teruel, "Compact multibeam fully metallic geodesic Luneburg lens antenna based on non-euclidean transformation optics," *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 12, pp. 7383–7388, Dec 2018.

[7] N. Fonseca, Q. Liao, and O. Quevedo-Teruel, "Compact parallel-plate waveguide half-Luneburg geodesic lens in the Ka-band," *IET Microwaves, Antennas & Propagation*, vol. 15, Dec 2020.

[8] O. Zetterstrom, R. Hamarneh, and O. Quevedo-Teruel, "Experimental validation of a metasurface Luneburg lens antenna implemented with glide-symmetric substrate-integrated-holes," *IEEE Antennas and Wireless Propagation Letters*, pp. 1–1, May 2021.

[9] C. A. Balanis, *Antenna Theory: Analysis and Design (4th Edition)*. Hoboken: John Wiley & Sons, 2016, pp. 25–126. [Online]. Available: https://app.knovel.com/hotlink/toc/id:kpATADE01N/antenna-theory-analysis/antenna-theory-analysis

[10] D. M. Pozar, *Microwave engineering, (4th Edition)*. Hoboken: N.J: Wiley, 2012, pp. 95–164.

[11] D. M. Pozar, *Microwave engineering, (4th Edition)*. Hoboken: N.J: Wiley, 2012, pp. 165–220.

[12] C. A. Balanis, *Antenna Theory: Analysis and Design (4th Edition)*. Hoboken: John Wiley & Sons, 2016, pp. 783–861. [Online]. Available: https://app.knovel.com/hotlink/toc/id:kpATADE01N/antenna-theory-analysis/antenna-theory-analysis

[13] C. A. Balanis, *Antenna Theory: Analysis and Design (4th Edition)*. Hoboken: John Wiley & Sons, 2016, pp. 719–755. [Online]. Available: https://app.knovel.com/hotlink/toc/id:kpATADE01N/antenna-theory-analysis/antenna-theory-analysis

[14] *RT/duroid® 6002 High Frequency Laminates*, (2021), May. Rogers Corporation, RT6002. [Online]. Available: https://www.rogerscorp.com/-/media/project/rogerscorp/documents/advanced-connectivity-solutions/english/data-sheets/rt-duroid-6002-laminate-data-sheet.pdf

[15] *SMA Connector*, (2021, May). RND Connect, RND 205-00498. [Online]. Available: https://www.elfa.se/en/sma-connector-socket-50ohm-straight-panel-mount-\\rnd-connect-rnd-205-00498/p/30061839

[16] M. T. Ali, Aizat, I. Pasya, M. H. Mazlan Zaharuddin, and N. Ya'acob, "E-shape microstrip patch antenna for wideband applications," in *2011 IEEE International RF Microwave Conference*, 2011, pp. 439–443.

[17] M. T. Islam, N. Misran, M. N. Shakib, and B. Yatim, "Wideband stacked microstrip patch antenna for wireless communication," *2008 IEEE International Symposium on Parallel and Distributed Processing with Applications*, pp. 547–550, Dec 2008.

3D-Printed Geodesic Reflective Luneburg Lens Antenna for X-Band

Lukas Jonasson and David Oxelmark

***Abstract*—With the rise of 5G and the increasing number of devices, novel antenna designs are needed to meet the demand of the future. In this report, the authors present a design and experimental verification of a 3D-printed Geodesic Modulated Reflective Luneburg lens antenna working at the X-Band, 8-12 GHz. The lens profile is calculated from the refractive index of a flat system using transformation optics. Furthermore, the lens is modulated to minimize the height and chamfers are implemented to reduce reflections. A sliding waveguide connected to a coaxial cable is used to excite the lens while the transmitted signal is radiated from a sinusoidal flare. A copper-lined PLA substrate constitutes the 3D-printed lens. The authors achieved a S_{11} below -10 dB across the spectrum and a realized gain exceeding 10 dB across the sweeping angles at 12 GHz, showcasing the usability as a directed antenna.**

***Sammanfattning*—Med det nya 5G nätverket och den ökande mängden enheter behövs nya antenner för att möta framtidens efterfrågan. I denna rapport presenterar författarna en design och experimentell verifiering av en 3D-printad geodesisk modulerad reflekterande Luneburg linsantenn i X-bandet, 8-12 GHz. Linsprofilen beräknas från brytningsindexet för ett platt system med transformationsoptik. Dessutom är linsen modulerad för att minimera höjden och kantavfasningar implementeras för att minska reflektioner. En glidande vågledare ansluten till en koaxialkabel används för att excitera linsen medan den sända signalen utstrålas från en vågledare med sinusformad avrundning. Ett kopparfodrat PLA-substrat utgör den 3D-printade linsen. Författarna uppnådde en S_{11} under -10 dB över spektrumet och en realiserad förstärkning överstigande 10 dB över svepvinklarna vid 12 GHz, vilket visar linsens användbarhet som riktad antenn.**

***Index Terms*—3D-printing, Geodesic lens, Reflective Luneburg lens, Transformation optics, Waveguide, X-band.**

Supervisors: Pilar Castillo-Tapia, Sarah Clendinning, Oscar Quevedo-Teruel

TRITA number: TRITA-EECS-EX-2021:170

I. Introduction

With the increasing amounts of connected devices, wirelessly transmitted data, and the use of new telecommunication systems, 5G and beyond, the antenna infrastructure has to expand. In addition, the new telecommunication systems add the use of millimeter wavelengths to increase the bandwidth, speed, and reliability. However, with higher frequencies the path loss increases. Further demands on novel antennas are thus introduced, such as having a high directivity combined with the ability to scan. Array antennas are a solution to increase directivity, but the required feeding networks increase in complexity with increasing frequency [1].

With the new demands, the long-ago introduced lens antennas prove to be of interest to revive and further develop. They operate similarly to regular optical lenses, e.g. glasses, and can either converge the light into a beam or diverge it depending on the focal point and shape. Size has been one of the limiting factors of the lens antennas, due to the reason that the lens has to be much larger than the wavelength. The use of millimeter wavelengths makes it possible to produce lens antennas of a reasonable size.

One of these lens antennas is the Luneburg lens, which was first introduced in the 1950's by Rudolph Luneburg. The Luneburg lens is a type of rotationally symmetrical graded index lens which can manipulate the path of rays. With a cylindrical source at the circumference, the Luneburg lens creates a plane wave at the antipode [2].

A Luneburg lens can be implemented in many ways. The discretization of a dielectric enables the design of a flat lens, but introduces losses in the dielectric. The implementation of a glide-symmetric meta-surface enables the design of a flat Luneburg lens consisting of two air-filled parallel plates. This eliminates the use of dielectric and therefore losses within them, as done by [3] and [4].

With the introduction of transformation optics, the same result of the Luneburg lens can be achieved with different geodesic shapes. This enables the lens to be manufactured with a parallel plate waveguide out of metal, with air in between, that conforms to a surface [5]. By creating a geodesic lens and modulating it to reduce its size, a high performance can be achieved, as done by [6] and [7]. In a new article, a fully metallic modulated geodesic Luneburg lens antenna was designed to be half rotationally symmetric, whilst maintaining similar performance of a fully rotational lens [8].

The reflective Luneburg lens (RLL) has the same characteristics and advantages as the Luneburg lens, but obtains a larger aperture by reflecting the wave to the outside of a lower waveguide. This is achieved by using a different refractive index profile to get the desired ray path [9]. In this paper, the authors propose a design for a 3D-printed geodesic RLL in the X-band to explore a possible solution to increase the aperture.

II. Theory

In this section is the fundamental theory needed to understand the work explained. In addition it is also described how the theory is applied.

A. Waveguide

A waveguide is a type of transmission line used to guide a propagating wave. The characteristics of a waveguide depend on its geometry, materials, and dimensions. This determines what type of waves can propagate within. In a waveguide

different modes of a wave can transverse. In a single-conductor waveguide, a transverse electromagnetic wave (TEM) cannot propagate. This is induced due to the boundary conditions at the sides where the tangential electric field is zero. However, transverse magnetic (TM) and transverse electric (TE) waves are able to propagate in these types of waveguides. TM and TE waves have a nonzero electric and magnetic field respectively and a nonexistent magnetic and electric field respectively in the propagation direction [10].

A rectangular waveguide is a type of single-conductor waveguide and consists of a width a and height b, with $a > b$. The cutoff frequency, where the propagation constant $\gamma = 0$, of the modes in a rectangular waveguide is determined by

$$(f_c)_{mn} = \frac{1}{2\sqrt{\mu\varepsilon}}\sqrt{\left(\frac{m}{a}\right)^2 + \left(\frac{n}{b}\right)^2}. \tag{1}$$

The dominant mode, the mode with the lowest cutoff frequency, of a rectangular is TE_{10} with cutoff frequency

$$(f_c)_{TE_{10}} = \frac{1}{2a\sqrt{\mu\varepsilon}}. \tag{2}$$

The E-field of different modes are illustrated in Fig. 1.

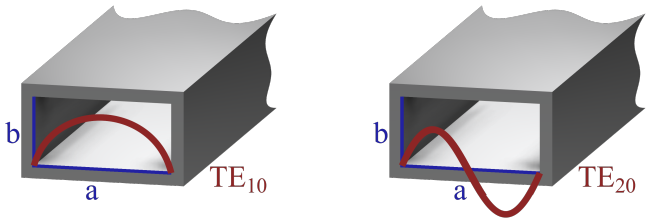


Fig. 1: Illustration of TE_{10} and TE_{20} with red sinusoidal as the amplitude of the E-field.

A parallel-plate waveguide (PPW) is a waveguide consisting of two parallel plates, separated with height b. The cutoff frequency of a PPW is determined by

$$f_c = \frac{n}{2b\sqrt{\mu\varepsilon}} \tag{3}$$

and the dominant mode is the TEM mode [10].

B. Scattering Parameters

The scattering parameters (S-parameters) describe the relationship between the voltage of the incident wave on a port and the voltage of the wave reflected by a port. If the incident voltage is expressed as V_i^+ on port i and the reflected voltage is expressed as V_i^- on port i the 2D scattering matrix is defined as

$$\begin{bmatrix} V_1^- \\ V_2^- \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} V_1^+ \\ V_2^+ \end{bmatrix}. \tag{4}$$

Therefore, the S-parameters can be calculated as

$$S_{ij} = \frac{V_i^-}{V_j^+} \tag{5}$$

where the incident voltages are set to 0 except V_j^+ [11]. The S-parameters are used to evaluate the amount of reflections, thus the performance of the designed parts.

C. Radiation pattern of an antenna

In order to determine the performance of an antenna, one needs to know how it radiates. The directivity, D, of an antenna is defined as

$$D = \frac{U_{max}}{U_{av}} = \frac{4\pi U_{max}}{P_r} \tag{6}$$

where U is the radiation intensity and P_r is the total time-averaged radiated power. Directivity is a dimensionless unit and is often expressed in dBi, referred to the radiation of an isotropic antenna. Associated with directivity is the beam width, which is a measure to describe the angular width of the main radiation lobe. This is defined by the angles between the half power, -3 dB, of the maximum. Closely related to directivity are also the sidelobe levels of an antenna, which describes the radiation intensity of the undesirable directions of radiation. To assess the efficiency of an antenna, a realized gain can be evaluated when taking losses into account. The antenna efficiency is then calculated as the realized gain divided by the simulated directivity [10].

D. Reflective Luneburg Lens

A lens antenna is similar to a regular lens, such as glasses, in the way that it bends the light in a controlled manner. The RLL is closely related to the regular Luneburg lens, but instead of creating a plane wave at the antipode, it reflects a plane wave to a lower waveguide. This is achieved by using the planar refractive index profile

$$n_{RLL} = n_0 \left(\frac{-1 + \sqrt{1 + 8(\rho/R)^2}}{2(\rho/R)^2}\right)^{3/2} \tag{7}$$

to acquire the desired ray path [9]. In Eq. (7) ρ/R is the normalized radius and n_0 is the refractive index at the periphery.

E. Transformation Optics

A lens with a known rotationally symmetric graded index can be converted to an arbitrary rotationally symmetric curved index. This is made possible by equating the ray paths on the surface of the lenses to achieve equivalent optical lengths. Firstly, two corresponding circles, S_1 and S_2, on the curves can be expressed by

$$n_1(r)2\pi r = n_2(\theta)2\pi R(\theta)sin(\theta) \tag{8}$$

followed by the two arc lengths, l_1 and l_2, of the two systems according to

$$n_1(r)dr = n_2(\theta)\sqrt{R(\theta)^2 + R'(\theta)^2}d\theta. \tag{9}$$

In Eq. (8) and Eq. (9), n_1 refers to the refractive index of the flat system while n_2 refers to the arbitrary system. r is the normalized radius of the systems and the equations are expressed in a cylindrical coordinate system [5]. An illustration of the two systems is presented in Fig. 2.

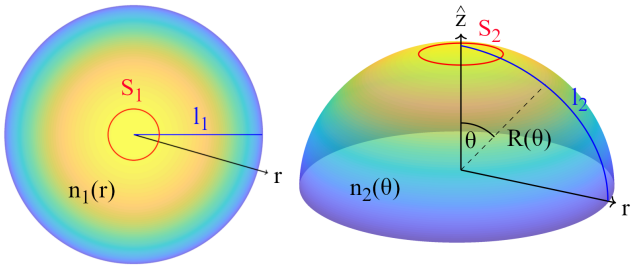


Fig. 2: Illustration of equated ray paths with arbitrary gradient refractive index. $\hat{z}$ and r are normalized.

III. System Design

In the project, the authors simulate the lens in CST Microwave Studio to evaluate the design. The simulations are setup with an accuracy of -40 dB with a perfect electric conductor (PEC) to contain the fields. In the following subsections are the design and simulation process of the system parts explained.

A. Lens Design

The goal of this project is to have a homogeneous lens with a refractive index of 1. This eliminates the need of dielectrics inside the lens, which would introduce losses. By solving the equation system of (8) and (9) numerically in matlab, with $n_2 = 1$ and n_1 as (7), a $R(\theta)$ can be found. The authors use the "vpa-solve" function built into MATLAB [12] and the Euler forward method [13]. The result are achieved with 10000 points with the boundary condition $r(10000) = 1$. $R(\theta)$ and θ could then be transformed to a cylindrical coordinate system via

$$r = R(\theta)sin(\theta) \tag{10}$$

$$\hat{z} = R(\theta)cos(\theta). \tag{11}$$

r and $\hat{z}$ are normalized such that $0 \leq r \leq 1$. The lens is rotationally symmetric around the z-axis and follows the calculated profile.

The lens in itself is frequency independent in theory, but in actuality, the lens needs to have a certain size to bend the rays. According to [9], the radius of the lens should at least be 2λ of the highest frequency. The authors decided to use 2.5λ for a lens to be designed in the X-band, 8-12 GHz. A lens radius of 93.75 mm and a lens height of 125.2132 mm are obtained. Fig. 3 shows the profile of the lens.

A 2D lens will not support the waves by itself. The rays need a curved PPW to travel through that follows the profile. In order to only excite the TEM mode in the PPW, the height should be less than $\lambda/4$ of the highest operating frequency [7]. A height of 6 mm is chosen, resulting in a 3 mm offset for each curve. The offset curves are determined by calculating the tangent and normal direction for each point of the lens profile and moving it in the direction of the normal ± 3 mm. Due to the discrete shape, the transformation introduces some wrong results at the ends, which are manually aligned with the axes.

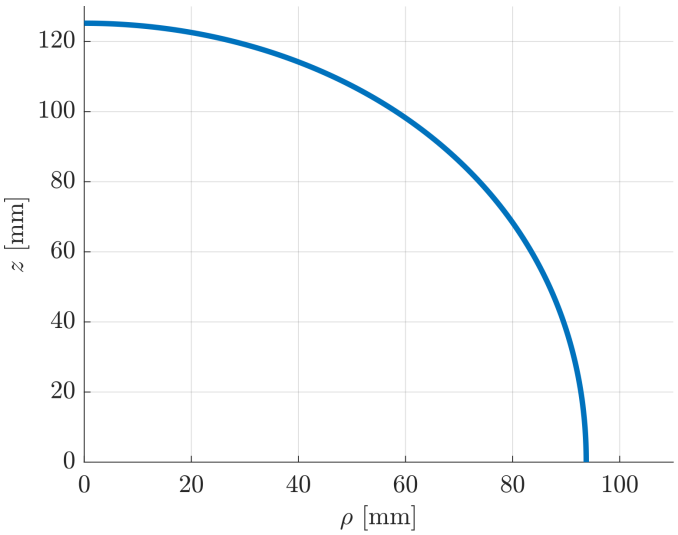


Fig. 3: Calculated lens profile.

Since the lens is quite tall it would be appropriate to minimize the height to decrease the overall footprint of the lens. This is achieved by modulating the lens, essentially mirroring the lens profile in a plane parallel to the XY-plane. Since the ray path of the lens is preserved, the refractive index will remain. However, the modulation introduces a singularity at the mirror point, which causes severe reflections [6]. To accommodate the lower PPW and the offset, the lens is modulated to minimize the overall height while keeping $z(0) \geq 3$ mm. The modulated lens with offset curves is illustrated in Fig. 4. The total height of the lens reduces by 48%.

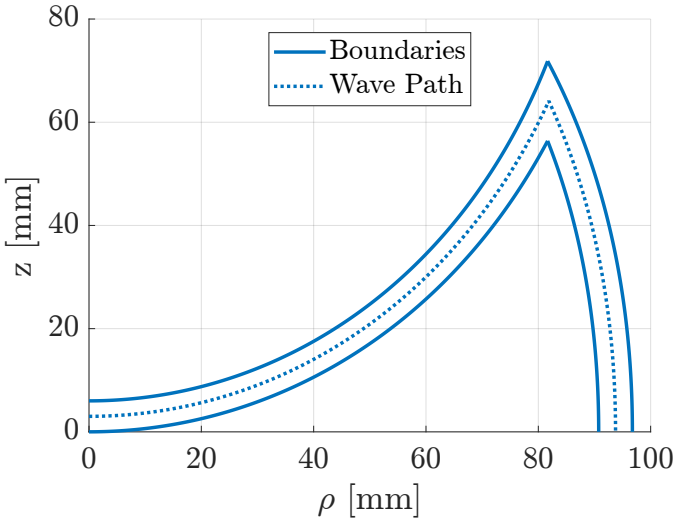


Fig. 4: Modulated lens with calculated offset curves.

B. Chamfer Design

In an attempt to minimize the reflections caused by the modulation, a chamfer is introduced at the modulating point. Another two chamfers are added to the design, one at the feeding point and one at the boundary between the lens and the

lower PPW. These chamfers are manually optimized to minimize S_{11} for the full system. Minimization of the reflections can also be accomplished via small radii, as performed in [7]. The chamfers minimize reflections but alter the shape of the curved PPW and hence the refractive index. With the change of refractive index, possible errors could occur. However, the footprint of the lens has been determined to be of greater importance to achieve a low profile lens. The chamfers are illustrated in Fig. 5.

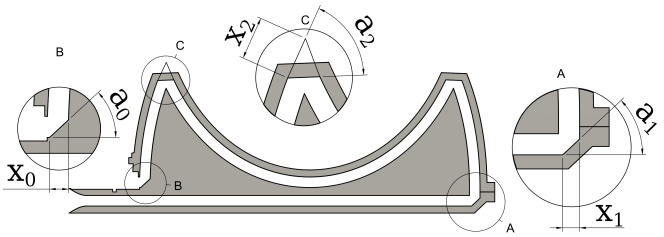


Fig. 5: Determined chamfers with angles and widths. The dimensions are stated in Table I.

TABLE I
DIMENSIONS OF THE CHAMFERS

Angle	Width
$a_0 = 43.5°$	$x_0 = 5.4\,mm$
$a_1 = 43°$	$x_1 = 4.8\,mm$
$a_2 = 66.25°$	$x_2 = 5\,mm$

C. Coaxial Cable to Waveguide Design

The lens requires excitation at the periphery of the PPW. This is solved by connecting a coaxial cable to a rectangular waveguide that only excites the first mode, TE_{10}. The authors use a coaxial cable connector manufactured by Distrelec [14]. The height of the waveguide at the periphery is determined by the height of the curved PPW, 6 mm, and the width of 24 mm is chosen to only excite the dominant mode. The dispersion diagram of the waveguide is shown in Fig. 6. The coaxial connector needs to be inserted in the waveguide and to obtain a better match a wider and taller waveguide is implemented. Therefore a transition between the two waveguides is needed. A two-step transition in both width and height simultaneously is implemented. Furthermore, the distance between the coaxial connector and the backshort, the back wall, (L_0) and the length of insertion of the coaxial cable is determined (H_0). All dimensions of the coaxial cable and waveguide are manually optimized to minimize S_{11} of the full system. The shape of the waveguide is illustrated in Fig. 7 with the dimensions in Table II. In the simulations, the waveguide is connected to a PPW to mimic the lens. The S_{11} of the coaxial connector and waveguide connected to a PPW can be seen in Fig. 7.

D. Flare Design

To create a transition between the lens and the environment a sinusoidal flare is attached. The flare is designed to enhance the direction pattern and increase the aperture of the lens. A

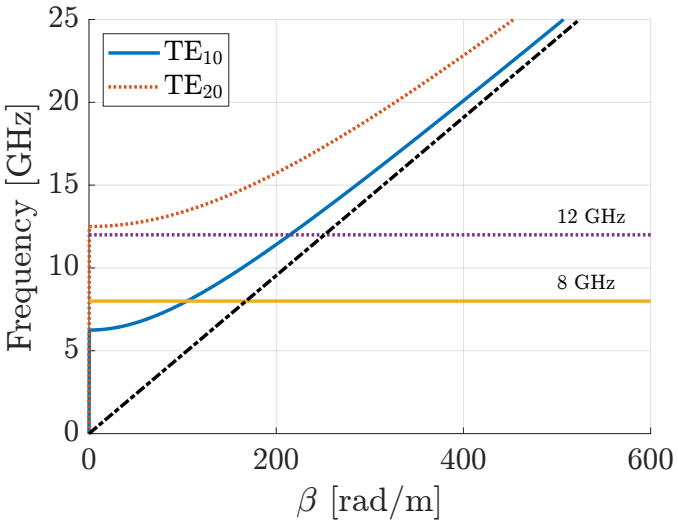


Fig. 6: Dispersion diagram of the waveguide. Illustrates the cutoff frequencies for the modes, where $\beta = 0$. Line of light is in black.

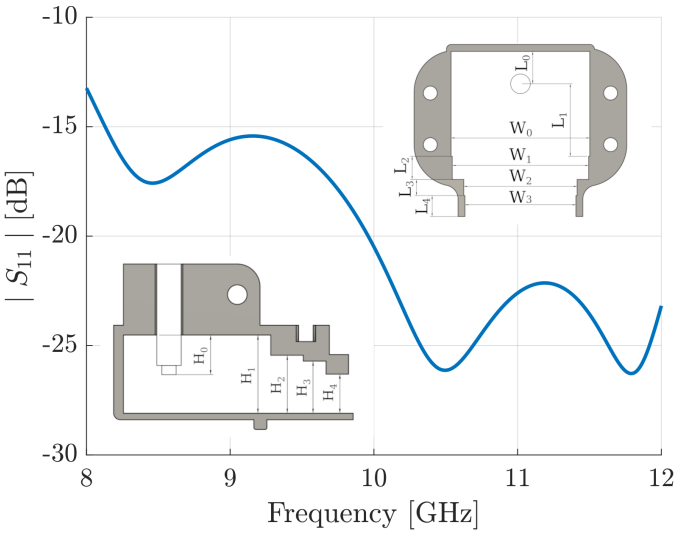


Fig. 7: S_{11} of the coaxial cable to waveguide with the dimensions stated in Table II.

TABLE II
DIMENSIONS OF THE WAVEGUIDE

Width	Height	Length
$W_0 = 30\,mm$	$H_0 = 7\,mm$	$L_0 = 7.35\,mm$
$W_1 = 29.5\,mm$	$H_1 = 12\,mm$	$L_1 = 15.3\,mm$
$W_2 = 24.5\,mm$	$H_2 = 8.9\,mm$	$L_2 = 5\,mm$
$W_3 = 24\,mm$	$H_3 = 8\,mm$	$L_3 = 3.5\,mm$
	$H_4 = 6\,mm$	$L_4 = 4.5\,mm$

scaled sinus defined in the interval $[-\frac{\pi}{2}, 0]$ outlines the shape of the flare and is optimized by manually changing the scaling of the sinus by altering the width and height. The length and the offset of the flare from the lens is adjusted to fit the waveguide on top of the flare. This is done to decrease the impact of the waveguide on the radiating wave, but to keep the size as small as possible it is adjusted to exactly fit the waveguide. The final shape and dimensions of the flare are illustrated in Fig. 8 and the S_{11} is displayed in Fig. 9.

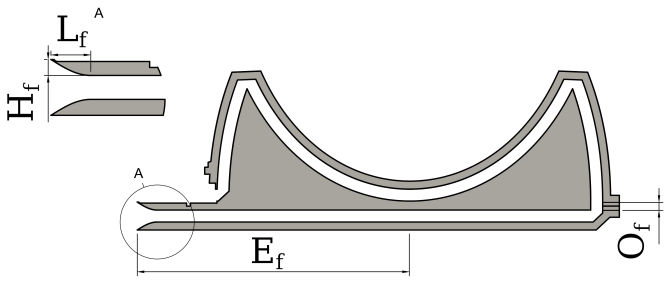


Fig. 8: Flare and lower PPW with dimensions stated in Table III. E_f is the extension of the flare and O_f the offset of the lower PPW. H_f is the amplitude and L_f the length of the quarter sinus curve.

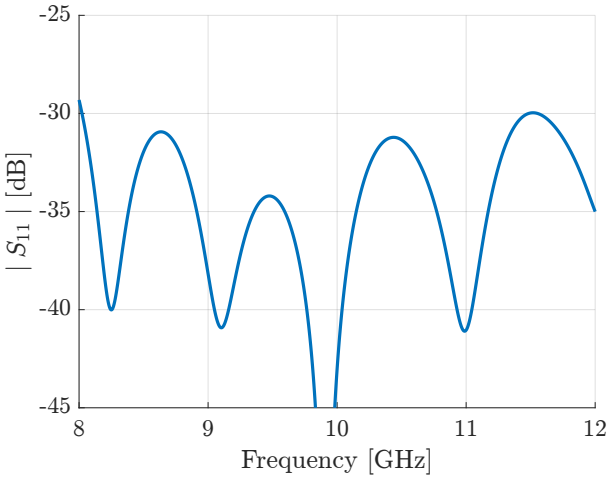


Fig. 9: S_{11} for the flare.

TABLE III
DIMENSIONS OF THE FLARE

Length	Height	Offset	Extension
$L_f = 10\,mm$	$H_f = 4\,mm$	$O_f = 4.5\,mm$	$E_f = 136.75\,mm$

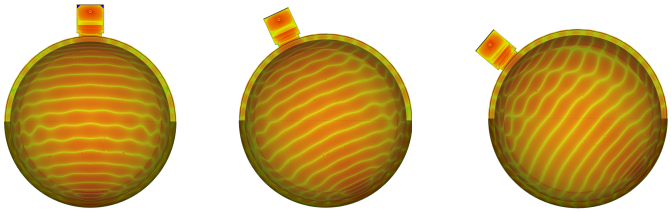
IV. Simulation Results

A. E-field

The goal of the lens is to achieve a high directivity from the transformation of a cylindrical wave to a plane wave. Fig. 10 shows the path of the rays through the lens, while Fig. 11 shows the simulated E-field distribution in the lower PPW that radiates out from the flare. Here it can be seen that the wave is not perfectly planar, but very close. The reason for this inconsistency has not been determined, but could be due to, alone or collectively, the chamfers, offset of the lower PPW or the extension of the flare. The E-fields also illustrate a problem with the design. When a_f is increased, the different chamfers become less effective or are obstructing the waves, due to the half-symmetric nature of the lens.

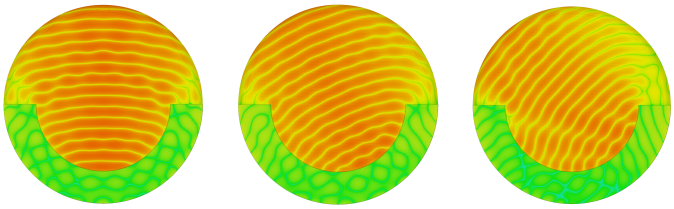
B. Simulated Directivity

Before manufacturing, the lens is simulated to perceive the performance. The simulated lens achieves a directivity above 10 dBi with low sidelobe levels. The simulated farfields are presented in Fig. 12 with accompanying results in Table. IV.



(a) $a_f = 0°$ (b) $a_f = 25°$ (c) $a_f = 50°$

Fig. 10: Simulated E-field distribution at 10 GHz in the geodesic lens for different feeding angles. Cross-sectional view with z-normal plane at 0.01 mm.



(a) $a_f = 0°$ (b) $a_f = 25°$ (c) $a_f = 50°$

Fig. 11: Simulated E-field distribution at 10 GHz in the lower PPW for different feeding angles. Cross-sectional view with z-normal plane at -10 mm.

V. Manufacturing

The lens is designed in Fusion 360 and is 3D-printed in blue and black PLA with a layer height of 0.3 mm. A copper tape is lined to the inside and acts as the curved PPW. The design consists of six parts, three for the lens and three for the waveguide.

Threaded inserts are inserted in the right and left part of the waveguide to assemble the coaxial cable connector with M2.5 screws. The waveguide is supposed to slide at the periphery of the lens, hence an arced pin is added to the bottom part. In the left and right part of the waveguide there is a hole to fit a M3 screw that fits into positive stops at 0°, 25° and 50° on the top part of the lens. The waveguide parts are printed with a varying wall thickness, with the thinnest at the bottom being 1 mm.

The manufactured lens consists of a bottom, middle, and top part, which are bolted together with M3 screws and bolts. The bottom and top part consist of the outer parts of the curved PPW, lower reflected PPW, and lower flare. This part is 4 mm thick. The middle part constitutes the upper reflected PPW,

TABLE IV
FARFIELD SIMULATIONS

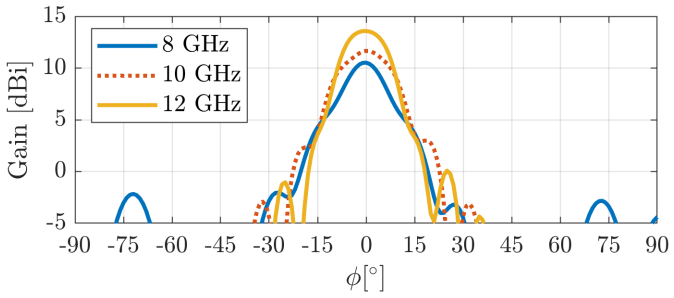
Freq.	a_f	Lobe Dir.	Gain	Sidelobes	Beam Width
8 GHz	0°	0°	10.5 dBi	−12.6 dB	14.9°
	25°	24°	10.9 dBi	−13.3 dB	14.0°
	50°	47°	11.2 dBi	−8.2 dB	13.8°
10 GHz	0°	0°	11.7 dBi	−8.7 dB	20.2°
	25°	25°	12.3 dBi	−16 dB	12.5°
	50°	48°	13.0 dBi	−8.7 dB	10.7°
12 GHz	0°	0°	13.6 dBi	−13.5 dB	16.1°
	25°	25°	14.2 dBi	−13.1 dB	12.7°
	50°	48°	14.5 dBi	−10.8 dB	9.8°

flare, and the inner part of the curved PPW. This part is printed with a 10% infill to minimize the weight. The three parts are put together using M3 bolts and nuts. To attach the middle part to the rest, an extra tab is printed at the flare, which fits in a matching grove on the other parts. A matching grove is implemented in the middle part on top of the upper reflected PPW to fit the arced pin on the waveguide. The rendered model is illustrated in Fig. 13 with a picture of the 3D-printed parts in Fig. 14.

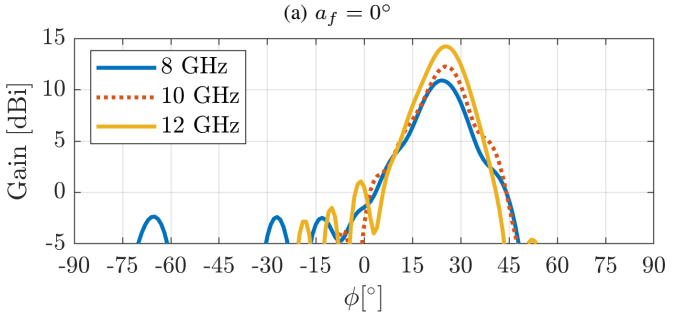
VI. Measurement Results

A. S-parameters

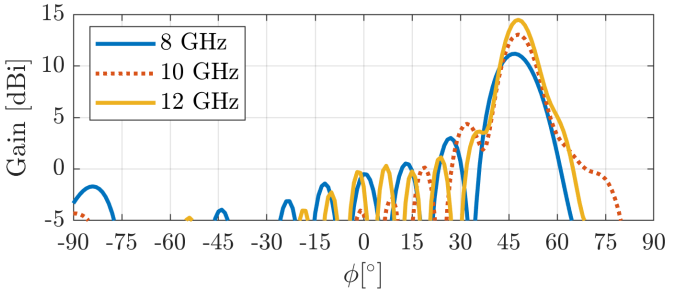
The authors performed the measurements of the lens at Royal Institute of Technology (KTH). A vector network analyzer (VNA) is used to measure the S_{11} parameters for the different angles. The measurement setup can be seen in Fig. 15, with the simulated and measured results presented in Fig. 16. The design achieved a S_{11} below -10 dB across the targeted frequency band for the different feeding angles, thus not having too much reflections and therefore transmitting the energy desirably.



(a) $a_f = 0°$

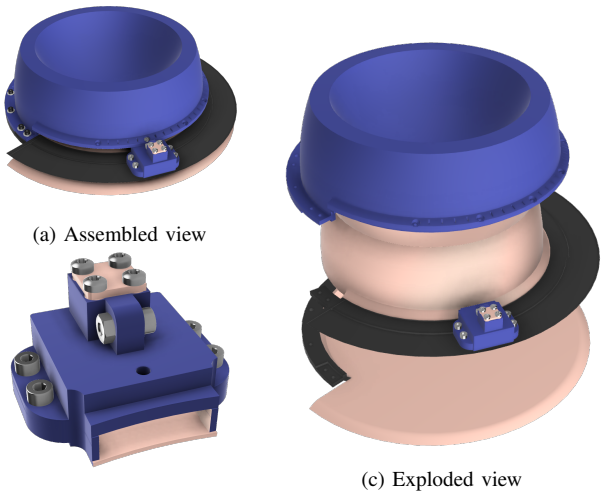


(b) $a_f = 25°$



(c) $a_f = 50°$

Fig. 12: Simulated gain of the lens presented in spherical coordinate system with $\theta = 90°$.

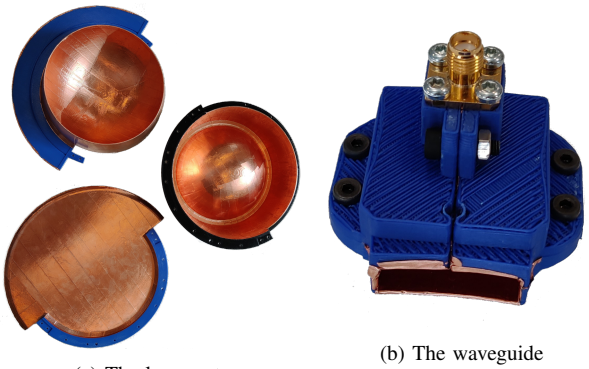


(a) Assembled view

(b) Assembled waveguide

(c) Exploded view

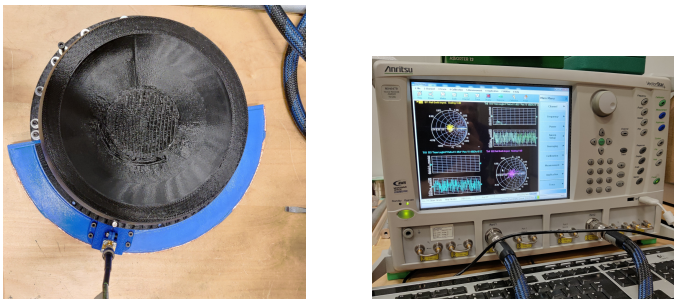
Fig. 13: Render of the CAD model.



(a) The lens parts

(b) The waveguide

Fig. 14: 3D-printed parts lined with copper tape



(a) The lens

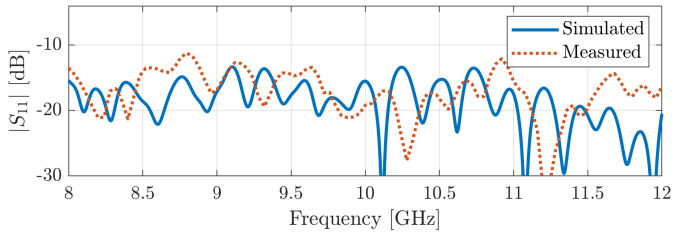
(b) VNA

Fig. 15: Measurement setup for the S-parameters.

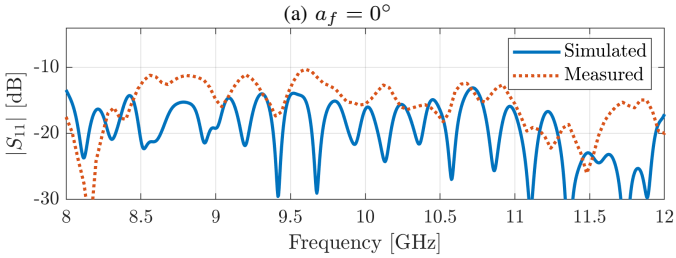
B. Farfields

The farfields of the lens were measured at KTH inside an anechoic chamber. To mount the lens in the chamber, two plastic rods were attached to the lens, as visualized in Fig. 17. Due to the flexibility in the 3D-printed PLA, the whole lens tilted and the flare became compressed. A mixture of masking and packing tape is used to straighten it up. The normalized directivity of the simulations and measurements are presented in Fig. 18.

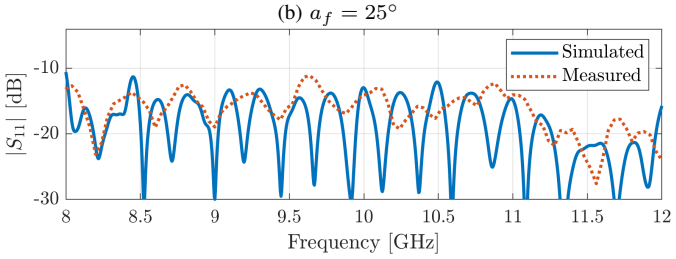
The measured directivity at 12 GHz matches the simulations



(a) $a_f = 0°$



(b) $a_f = 25°$



(c) $a_f = 50°$

Fig. 16: Simulated and measured S_{11} of the full system.

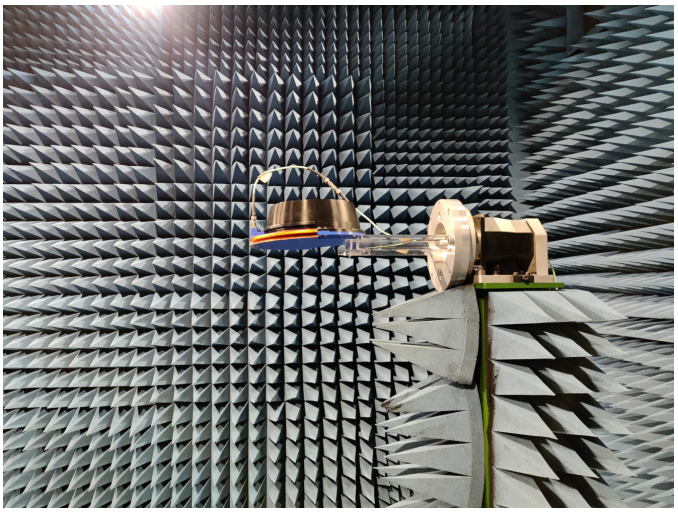
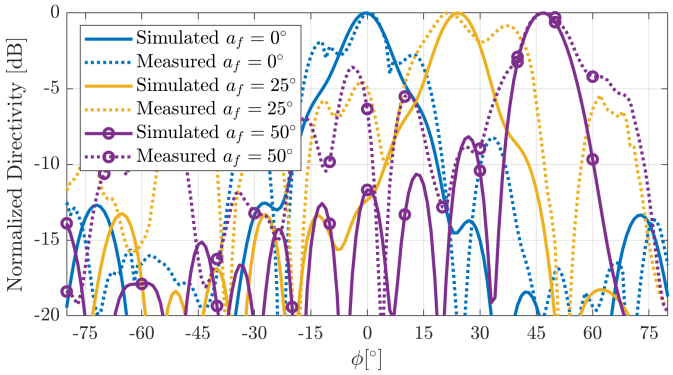


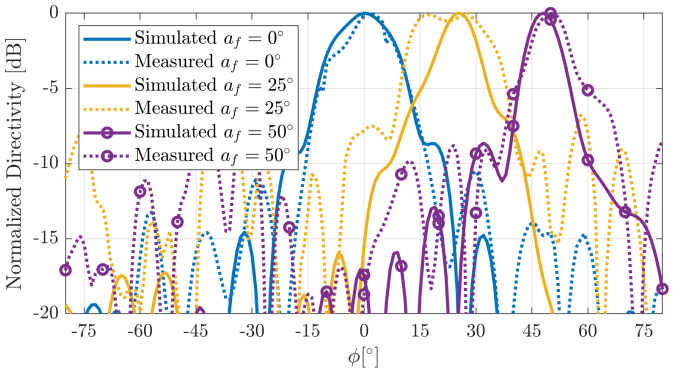
Fig. 17: The lens mounted in the anechoic chamber at KTH.

very closely. However, at 8 and 10 GHz the results are not as close. At 8 GHz, the lens barely achieves a usable directivity, with high sidelobe levels and wide beamwidth. At 10 GHz, the results are more varying. At $a_f = 0°$ the lens achieves sidelobes below -10 dB and matches the simulations. For $a_f = 25°$, the beamwidth is significantly larger with higher side lobes than the simulations. The side lobes are also higher with $a_f = 50°$. The calculated farfield values can be seen in Table V.

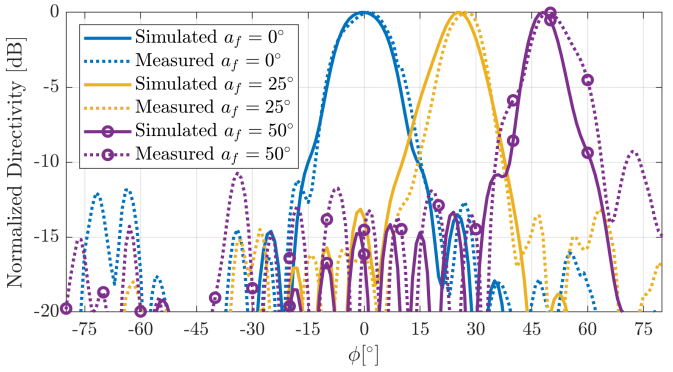
The measurement results are consequently not on par with the simulations, which could be correlated with the manufacturing. As mentioned before, the lens deformed during the measuring, leading to a possible change in the wave path. When manufactured, the heavy middle part of the lens was connected with 2 screws at each side. The deformation could be avoided by adding additional support on the outside of the lens, increasing the mechanical stability of the system. In addition, the PPW was constructed with copper tape which created small inconsistencies on the surface. This is not optimal and could have affected the final results. Another possible solution to prevent both of these problems is to manufacture the full system in aluminum.



(a) 8 GHz



(b) 10 GHz

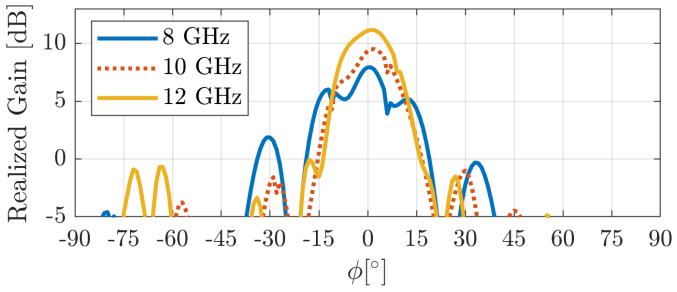


(c) 12 GHz

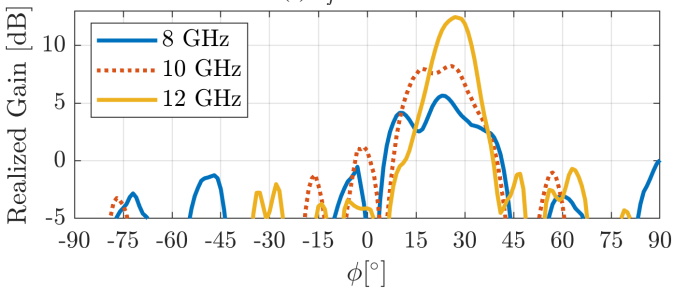
Fig. 18: Normalized farfields for different feeding angles. Presented in a spherical coordinate system with $\theta = 90°$.

To obtain the realized gain of the antenna, the gain of a reference antenna is needed. This was supplied by KTH for the desired frequency band. The measurements of the lens can then be scaled to the reference antenna. The realized gain is shown in Fig. 19. These results clearly show that the

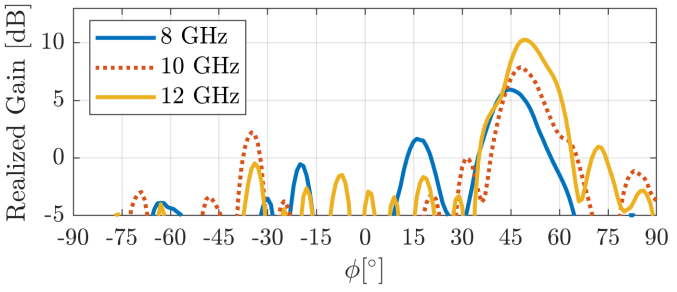
antenna functions at 12 GHz, with a realized gain above 10 dB. The other frequencies, as described earlier, have varying results. From the realized gain the antenna efficiency can be calculated. The lens is quite effective and achieves a radiation efficiency between 54.3% and 82.3% across the spectrum and feeding angles, as can be seen in Table VI.



(a) $a_f = 0^\circ$



(b) $a_f = 25^\circ$



(c) $a_f = 50^\circ$

Fig. 19: Realized gain of the lens presented in spherical coordinate system with $\theta = 90^\circ$.

TABLE V
FARFIELD MEASURMENTS

Freq.	a_f	Lobe Dir.	R. Gain	Sidelobe	Beam Width
8 GHz	0°	0°	7.9 dB	−1.9 dB	20.7°
	25°	22°	5.6 dB	−0.85 dB	30.6°
	50°	47°	5.9 dB	−3.6 dB	16.4°
10 GHz	0°	2°	9.5 dB	−1.4 dB	18.4°
	25°	28°	8.2 dB	−0.1 dB	25.2°
	50°	49°	7.9 dB	−9.3 dB	13.8°
12 GHz	0°	1°	11.2 dB	−11.2 dB	17.2°
	25°	27°	12.5 dB	−13.1 dB	11.7°
	50°	49°	10.3 dB	−9.3 dB	13.8°

VII. Conclusion

In this report, the authors presented a design of a 3D-printed Reflective Luneburg lens antenna for the X-band. The authors have shown that a geodesic reflective Luneburg lens is feasible and can be manufactured using a 3D-printer. The design showcases the possibilities of the novel lens at higher frequencies.

TABLE VI
ANTENNA EFFICIENCY

Freq.	a_f	Efficiency
8 GHz	0°	74.1%
	25°	54.3%
	50°	54.3%
10 GHz	0°	77.6%
	25°	62.4%
	50°	55.6%
12 GHz	0°	75.9%
	25°	82.3%
	50°	61.7%

VIII. Future Work

To build upon the presented design, the lens could be coupled with a leaky-wave antenna instead of the lower PPW to disperse the radiation, similar to what's done in [9]. This could drastically increase the directivity with the implementation of pin needle beam steering. To address the half-rotational nature of this lens, another transformation of the lens could be applied. The lens could be made fully rotationally symmetric by moving the feeding to the inside of the lens instead of at the periphery, as presented in [9]. It would also be of interest to manufacture the lens from solid metal to investigate the differences between the two manufacturing processes. Furthermore, the chamfers could be improved, either by better optimization of both the S-parameter and the directivity or by adapting a different way to minimize the reflections. The lens should also be designed for higher frequencies to be better adapted to the telecommunication industry.

Acknowledgment

The authors would like to thank the project supervisors Pilar Castillo-Tapia and Sarah Clendinning for their help and support during the project. The authors would also like to thank Oscar Quevedo-Teruel and the Division of Electromagnetic Engineering at KTH for the use of measurement equipment and providing financial support of the lens. Lastly the authors would like to thank the Student Workshop at KTH for allowing the use of their 3D-printers.

References

[1] Y. Wang, J. Li, L. Huang, Y. Jing, A. Georgakopoulos, and P. Demestichas, "5G Mobile: Spectrum Broadening to Higher-Frequency Bands to Support High Data Rates," *IEEE Vehicular Technology Magazine*, vol. 9, no. 3, pp. 39–46, Sept 2014.

[2] R. Luneburg, *Mathematical Theory of Optics*, 1st ed. Providence, RI: Brown University Press, 1944.

[3] O. Quevedo-Teruel, J. Miao, M. Mattsson, A. Algaba-Brazalez, M. Johansson, and L. Manholm, "Glide-Symmetric Fully Metallic Luneburg Lens for 5G Communications at Ka-Band," *IEEE Antennas and Wireless Propagation Letters*, vol. 17, no. 9, pp. 1588–1592, Sept 2018.

[4] O. Zetterstrom, R. Hamarneh, and O. Quevedo-Teruel, "Experimental Validation of a Metasurface Luneburg Lens Antenna Implemented with Glide-Symmetric Substrate-Integrated-Holes," *IEEE Antennas and Wireless Propagation Letters*, vol. 20, no. 5, pp. 698–702, Feb 2021.

[5] R. C. Mitchell-Thomas, O. Quevedo-Teruel, T. M. McManus, S. A. R. Horsley, and Y. Hao, "Lenses on Curved Surfaces," *Opt. Lett.*, vol. 39, no. 12, pp. 3551–3554, Jun 2014.
[6] N. J. G. Fonseca, Q. Liao, and O. Quevedo-Teruel, "Equivalent Planar Lens Ray-Tracing Model to Design Modulated Geodesic Lenses Using Non-Euclidean Transformation Optics," *IEEE Transactions on Antennas and Propagation*, vol. 68, no. 5, pp. 3410–3422, May 2020.
[7] Q. Liao, N. J. G. Fonseca, and O. Quevedo-Teruel, "Compact Multi-beam Fully Metallic Geodesic Luneburg Lens Antenna Based on Non-Euclidean Transformation Optics," *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 12, pp. 7383–7388, Dec 2018.
[8] N. J. Fonseca, Q. Liao, and O. Quevedo-Teruel, "Compact Parallel-Plate Waveguide Half-Luneburg Geodesic Lens in the Ka-Band," *IET Microwaves, Antennas & Propagation*, vol. 15, no. 2, pp. 123–130, Apr 2020.
[9] J. Ruiz-García, E. Martini, C. D. Giovampaola, D. González-Ovejero, and S. Maci, "Reflecting Luneburg Lenses," *IEEE Transactions on Antennas and Propagation*, submitted for publication.
[10] D. Cheng, *Field and Wave Electromagnetics*, 2nd ed., ser. Addison-Wesley series in electrical engineering. Boston: Addison-Wesley, 1989, pp. 534–554,610–611.
[11] D. Pozar, *Microwave Engineering*, 4th ed. Hoboken, NJ: Wiley, 2011, pp. 178–179.
[12] Matlab. Vpa-Solve. Mathworks. Accessed: 2021-04-29. [Online]. Available: https://www.mathworks.com/help/symbolic/sym.vpasolve.html
[13] T. Sauer, *Numerical Analysis*, 2nd ed. Boston,MA: Pearson, 2012, pp. 283–290.
[14] Distrelec, *SMA Connector*. [Online]. Available: https://www.elfa.se/Web/Downloads/_t/ds/rnd_205-00498_eng_tds.pdf

Glide-symmetric Holey EBG Filter Using Multiple Unit Cell Designs

Gustav Eliasson and Lucas Åkerstedt

***Abstract*—There are more connected wireless devices than ever before and with the rise of new smart systems such as self-driving cars and smart cities new antenna solutions for transmitting signals are needed. One important part of these systems is the filters which filter out all the unwanted signals. In this report, we present a solution for manufacturing such a filter with a passband from 26-29 GHz and a stopband from 29-60 GHz using a fully metallic glide-symmetric structure. Ideas of combining multiple unit cell designs to achieve wider stopbands and higher attenuation are explored using dispersion engineering where the advantages and the disadvantages of using this method are presented. Furthermore, ways of combining the filter to standard connections using a coaxial cable to waveguide transition are proposed and designed. The usage of multiple unit cell designs is proven to be a solution for achieving wider stopbands with minimum coupling between modes.**

***Sammanfattning*—Det finns fler trådlösa enheter uppkopplade än någonsin tidigare och med ökningen av nya smarta system som självkörande bilar och smarta städer finns ett behov av nya antennlösningar för överföring av information. En viktig del av dessa system är filtren som filtrerar bort alla oönskade signaler. I denna rapport presenterar vi en lösning för att konstruera ett sådant filter med ett passband från 26-29 GHz och ett stoppband från 29-60 GHz med en helt metallisk glidsymmetrisk struktur. Idéer att kombinera flera enhetscellsdesigner för att uppnå bredare stoppband och högre attenuering undersöks med hjälp av dispersionsteknik, där fördelarna och nackdelarna med att använda denna metod presenteras. Dessutom föreslås och utformas sätt att kombinera filtret till standardanslutningar med en koaxialkabel till vågledarövergång. Användningen av flera enhetscell designer visar sig vara en lösning för att skapa breda stoppband med minimal koppling mellan ”modes”.**

***Index Terms*—Glide symmetry, waveguide filter, 5G, Electric band gap, unit cell, coaxial cable, holey structures, mm-wave**

Supervisors: Pilar Castillo Tapia, Freysteinn Vidar Vidarsson and Oscar Quevedo Teruel

TRITA number: TRITA-EECS-EX-2021:171

I. Introduction

DUE to a drastic increase in wireless devices the currently used frequency band is continuously getting crowded. Antenna solutions for higher frequency bands, such as the millimeter wave (mm-wave) frequency range, are therefore needed [1]. However, one problem that arises in the mm-wave frequency range is that conventional discrete component-circuits are no longer suitable [2]. Discrete components will introduce high losses in the mm-wave frequency range as a result of their dielectric structure [2]. Therefore lossless ways of filtering signals in the mm-wave regime are needed. One possible solution is traditional waveguide filters which have a very high Q-factor, very low leakage and low insertion loss, because of their metallic structure [3]. An example of such a filter would be the waffle iron filter, as seen in [3]. However, traditional waveguide filters require a much more intricate manufacturing process as a result of metallic pins [3].

Using a rather new technology called ”glide-symmetric holey EBG (electromagnetic band gap) structure”, a waveguide filter can be implemented and designed for the mm-wave frequency range [2] [4]. This technology relies on holes and their placement in a metallic structure, as can be seen in [5], to produce the desired filtering capabilities. Therefore the use of metallic pins can be avoided, and implementation of these holes can be used to achieve similar results as described in [6]. These glide-symmetric holes have also been proposed to reduce leakage in between flanges [7] and to design low dispersive leaky wave antennas [8].

In addition to its filtering abilities, this technology can be used to manufacture waveguide filters at a much lower cost than its adversaries as a result of its simplistic design [9].

In this report, a waveguide filter is developed using this glide-symmetric holey EBG-technology. The developed filter stands out from its relatives because of a slight change of the glide-symmetric unit cell. Instead of using the commonly used unit cell design, the unit cell structure has been rotated 45 degrees to create a new unit cell that can provide a wider stopband, which is proposed in [5]. The filter provides a passband from 26-29 GHz and a stopband from 29-60 GHz.

II. Theory

A. Periodic structures

A periodic structure is a structure in which different symmetry operations can be performed without changing the structure. Depending on these symmetry operations the lattice structure is categorized into different lattice groups where this project will focus on the Bravais 2D square lattices. From this lattice structure one can define primitive lattice vectors, where any integer combination of these vectors will describe a point in the lattice as seen in Fig. 1a). To describe the periodicity of this lattice a Fourier transformation can be performed resulting in what is called a reciprocal lattice. This structure has its own primitive vectors and they are defined by the following equation

$$\vec{T_1} = 2\pi \frac{\vec{t_1}}{a \mid \vec{t_1} \mid} \quad \vec{T_2} = 2\pi \frac{\vec{t_2}}{a \mid \vec{t_2} \mid} \tag{1}$$

where t_1, t_2 are the primitive physical lattice vectors and T_1, T_2 are primitive reciprocal lattice vectors seen in Fig. 1

b). Here the vectors have a length of $2\pi/a$ where a is the periodicity of the structure in the direction of the vector. These vectors are also grating vectors and are closely connected to the wave vectors, k. This is why it is preferable to study the structure in reciprocal space.

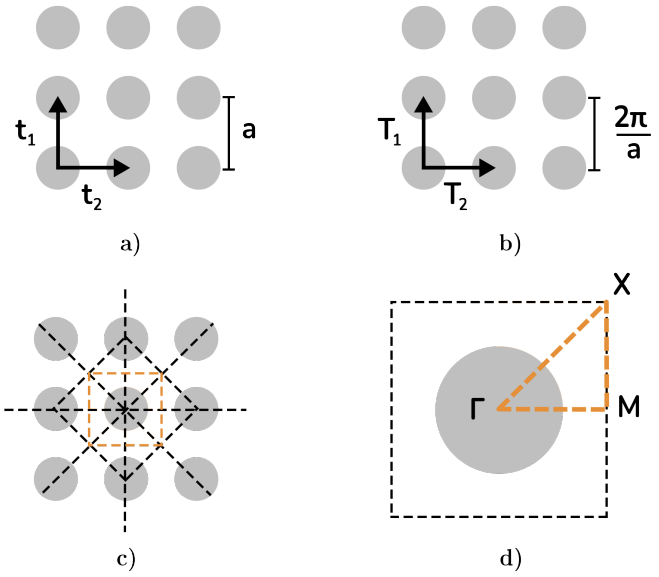


Fig. 1: a) the primitive physical lattice vectors for a 2D square lattice, b) the primitive reciprocal lattice vectors for a 2D square lattice, c) demonstrating the method of creating the Brillouin zone d) the Brillouin zone marked with the irreducible Brillouin zone in yellow.

B. Primitive cell

Primitive unit cells are the smallest volume of space that can be used to correctly reproduce the entire lattice. One method of construction a primitive unit cell is the Wigner-Seitz method. It is defined as the volume of space around a point in the lattice that is closer to that point than any other point in the lattice.

The same can be done on the reciprocal lattice where it's called the Brillouin zone. This zone can be constructed by using the already defined reciprocal lattice vectors which can be seen for a Bravais 2D square lattice in Fig. 1 c) where the yellow colored area is the Brillouin zone.

C. Bloch Theorem

Bloch theorem states that one only needs to study the Brillouin zone in order to characterise the electromagnetic properties of the whole structure. This is due to the fact that the waves in a periodic structure can be described as the sum of Bloch waves with the following appearance

$$E(r) = U(r)e^{jkr} \tag{2}$$

What this equation boils down to is some plane wave with the propagation constant k modulated by a function $U(r)$ which will have the same periodicity as the structure itself. This means that $U(k) = U(k+K)$ where K is a reciprocal lattice vector and that one only needs to study the Brillouin zone which is K by K large in the case of the 2D square lattice.

Within the Brillouin zone there exist more symmetries that could be exploited to reduce the computational effort even further which can be seen in Fig. 1 d). This yellow colored zone is what is called the irreducible Brillouin zone and in the case of the Bravais 2D square lattice it has 3 key points of symmetry, Γ, X, M. This is now the only zone that needs to be studied.

D. Glide symmetry

Glide symmetry is a form of symmetry. A 2D-periodic structure can be considered glide-symmetric if it has been translated half a period in two orthogonal directions followed by a mirroring. This can be expressed mathematically as:

$$G_{2D} = \begin{cases} x \to x + \frac{a}{2} \\ y \to -y \\ z \to z + \frac{a}{2} \end{cases} \tag{3}$$

where a is the periodicity of the periodic structure. An example of achieving a glide-symmetric structure can be seen in Fig. 2 where 2 a) displays a mirrored periodic structure, 2 b) displays the structure achieving glide symmetry in one dimension and 2 c) displays the structure achieving glide symmetry in two dimensions.

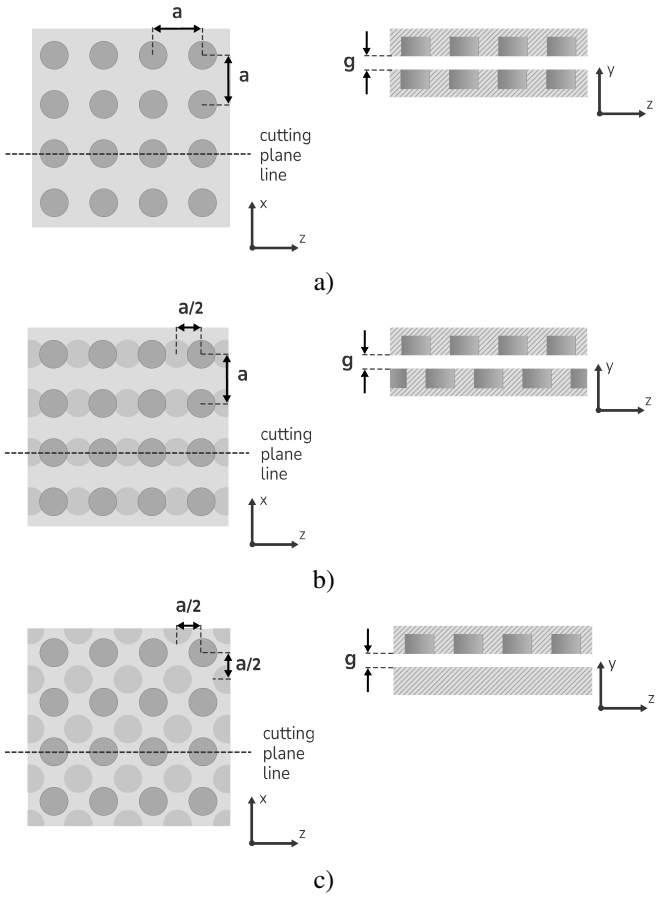


Fig. 2: Two periodic structures consisting of a plate with holes, where in a) the bottom plate is not shifted, b) the bottom plate is shifted $a/2$ in the z-direction and c) the bottom plate is shifted $a/2$ in z-direction and x-direction.

To accurately characterize the properties of a glide-symmetric structure one only needs to study the unit cell, or more specifically the Brillouin zone of the unit cell, of the structure as previously explained in II-C. Using the theory discussed in II-B a glide-symmetric structure can be reproduced from one unit cell as shown in Fig. 3. Thus the unit cells electromagnetic properties will determine that one of the whole structure.

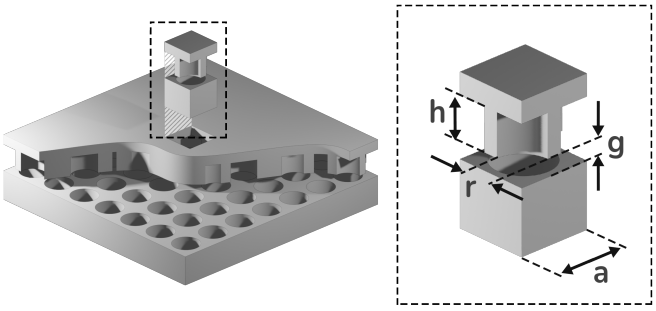


Fig. 3: Cut through glide-symmetric structure with one unit cell cut out, h is the hole depth, r is hole radius, g is the gap height between the plates and a is the periodicity of the structure.

E. Rectangular waveguide

A rectangular waveguide is a guiding structure that can effectively support the propagation of certain waves at certain frequencies. To support the propagating waves with minimal losses, the guiding structure needs to consist of a material with high conductivity. Therefore waveguides are often modeled with a perfect electric conductor (PEC) material. In practice PEC material is not achievable, hence most waveguides are made of metals with high conductivity.

In a rectangular waveguide, propagating waves can take on two major appearances: a transverse magnetic (TM) wave or a transverse electric (TE) wave. The TM mode consists of a transverse magnetic field and a longitudinal electric field whereas the TE mode consists of a transverse electric field and a longitudinal magnetic field. These modes have characteristic cutoff frequencies defined by the dimensions of the waveguide. When a mode can not propagate we call them evanescent modes which are modes that can not deliver power.

These cutoff frequencies for each mode can be calculated using this formula [10]:

$$\gamma = \sqrt{-\omega^2\varepsilon\mu + (\frac{m\pi}{b})^2 + (\frac{n\pi}{a})^2} \tag{4}$$

where a and b are our waveguide dimensions and n and m define the order of the mode. When plotting the propagation constant γ against frequency we get a dispersion diagram, for example the TE_{10} ($m = 1, n = 0$) mode shown in Fig. 4.

As previously mentioned there can exist two major modes in a rectangular waveguide: The TE mode and the TM mode. These modes all have a transverse field that will take on different appearances depending on the frequency of the propagating wave. The two components making up the transverse electric field of the TE mode can be calculated analytically with [10]:

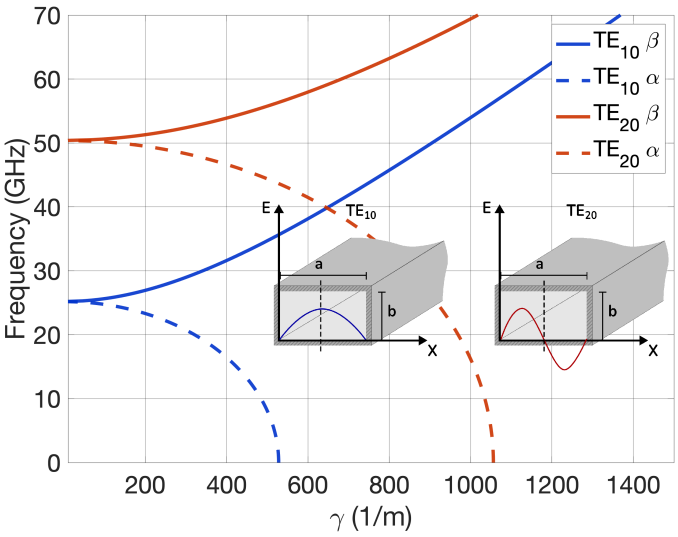


Fig. 4: Dispersion diagram of a rectangular waveguide with dimensions a = 5.95 mm and b = 2 mm. The TE_{10} mode is plotted in blue and the TE_{20} mode is plotted in red. β is the phase constant and α is the attenuation constant.

$$E_x(x,y) = \frac{j\omega\mu}{h^2}(\frac{n\pi}{b})H_0\cos(\frac{m\pi}{a}x)\sin(\frac{n\pi}{a}y) \tag{5}$$

and for the y-component of the transverse field

$$E_y(x,y) = -\frac{j\omega\mu}{h^2}(\frac{m\pi}{a})H_0\sin(\frac{m\pi}{a}x)\cos(\frac{n\pi}{a}y) \tag{6}$$

where

$$h^2 = (\frac{m\pi}{a})^2 + (\frac{n\pi}{b})^2 \tag{7}$$

In these formulas H_0 is the amplitude of the z-component of the magnetic field. From Equations 5 and 6 it is clear that for the TE_{10} mode the transverse electric field will only have a y-component, since $m = 1, n = 0$. With $a > b$ this means that the TE_{10} is the first TE mode to be excited in a waveguide. However, if the TE mode or TM mode has the lowest cutoff frequency remains to be examined.

The two components making up the transverse magnetic field of the TM mode can be calculated analytically with:

$$H_x(x,y) = \frac{j\omega\varepsilon}{h^2}(\frac{n\pi}{b})E_0\sin(\frac{m\pi}{a}x)\cos(\frac{n\pi}{a}y) \tag{8}$$

and for the y-component of the transverse field

$$H_y(x,y) = -\frac{j\omega\varepsilon}{h^2}(\frac{m\pi}{a})E_0\cos(\frac{m\pi}{a}x)\sin(\frac{n\pi}{a}y) \tag{9}$$

Where E_0 is the amplitude of the z-component of the electric field. With the transverse field equations for the TM mode, Equations 8 and 9, we can see that there is no way to have a transverse magnetic field with either $m = 1, n = 0$ or $m = 0, n = 1$. Thus the first TM mode to exist in a waveguide is the TM_{11} mode. Hence we have shown that the TE_{10} is the first mode to exist in a waveguide. This mode is referred to as the fundamental mode of the waveguide.

Now we only consider the TE mode and the y-component of the transverse electric field. Increasing the mode determining variable m, the transverse field will take on different appearances as can be seen in Fig. 5. More specifically, the different TE modes can be determined to be either odd or even. A mode that is even will have an E-field maximum in the center of the waveguide in the x-direction. A mode that is odd will have an E-field minimum in the middle of the waveguide in the x-direction, as explained in Fig. 5.

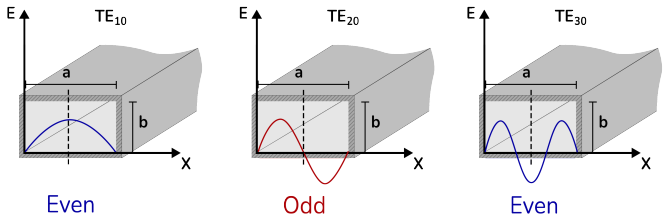


Fig. 5: The y-component of the transverse field for TE_{10}, TE_{20} and TE_{30} mode in a rectangular waveguide.

F. Dispersion diagram

When constructing filters one is often interested in the dispersion diagram of the unit cell used to design the filter. The dispersion diagram relates the frequency to the propagation constant of the modes supported in the structure.

When studying periodic structures in two dimensions the dispersion diagram will be three-dimensional since the wave can travel in an additional direction compared to the case of the waveguide. However, as mentioned earlier the dispersion diagram can be attained by only studying the irreducible Brillouin zone. Moreover, the extreme values of the dispersion relation in most cases occur at the boundary of the irreducible Brillouin zone and because of this, it is only necessary to sweep the propagation constant only over this boundary. By doing so the dispersion diagram can be plotted as a 2D graph seen in Fig. 6. This diagram shows multiple directions marked along the axis and shown on the unit cell. Where the Γ to X part of the graph shows the dispersion of waves traveling along the line ΓX marked on the unit cell. Γ to X means that the x-component of the phase constant is swept from [0, 180] degrees and the Γ to M will sweep the y-component of the phase constant from [0,180] degrees and the M to Γ will sweep both the x- and y-component from [180,0] of the phase constant.

G. Coaxial transmission line

A coaxial transmission line is a structure consisting of an inner conductor shielded by an outer one as seen in 7. This outer conductor is kept at ground potential and the signal is transmitted using the inner one. This ensures very little interference from electric and magnetic fields outside the transmission line. The dominant mode of a coaxial cable is the transverse electromagnetic (TEM) mode and it has no cutoff as demonstrated in the dispersion diagram in Fig. 7. An important factor for this project is the characteristic impedance of the coaxial cable, calculated with the following formula 10.

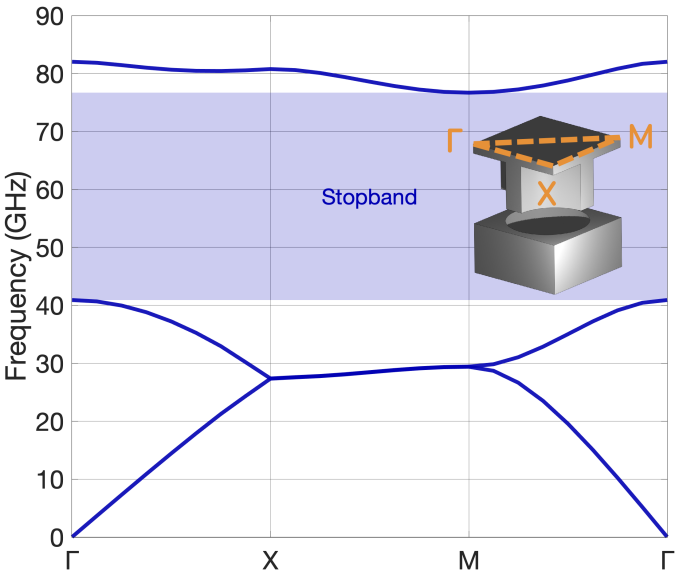


Fig. 6: Dispersion diagram for multiple directions with holey glide-symmetric unit cell marked with the computational zone.

$$Z_0 = \frac{1}{2\pi}\sqrt{\frac{\mu}{\varepsilon}}\ln\left(\frac{D_{out}}{D_{in}}\right) \tag{10}$$

where D_{out} is the outer conductor diameter and D_{in} is the inner conductors diameter.

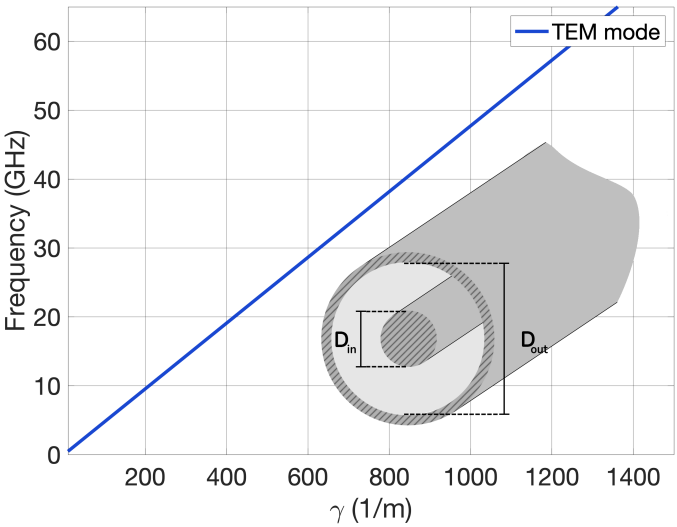


Fig. 7: Dispersion diagram of a coaxial transmission line with dimensions $D_{in} = 0.86$ mm and $D_{out} = 2$.

H. Scattering parameters

Scattering parameters are used to characterize the electromagnetic properties of many systems. This is done by placing ports and studying how much power goes through every port in relation to each other. In the case of a two port network an S-parameter matrix can be defined as seen in the following equation system:

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{22} & S_{21} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{11}$$

where the second number of the index in the S-parameters defines which port the energy or power is coming from and the first number defines at which port it exits. This means that the S_{11} will represent the amount of energy that is reflected back to port one from port one. The S_{21} demonstrates the amount of energy that that exits out of port 2 from port 1. The S-parameters are frequency dependent and will therefore always be calculated for a specified frequency and often plotted over the frequency range of interest. From these four parameters many others can be calculated such as the impedance, Z_0, or propagation constant, γ. For filter design it is however most interesting to see how much energy at different frequencies can pass through the structure.

III. System Design

The filter in its entirety can be seen in Fig. 8 where all the individual parts are listed. These parts are a coaxial cable to waveguide transition, a stepped impedance transformer, a tapering structure and a filtering structure consisting of three unit cell structures. In order to integrate the filter into any system the filter has to be designed with standard connections, thus the filter has coaxial connections at each end. The filter is designed to have a passband from 26-29 GHz and a stopband from 29-60 GHz, with an insertion loss of 0.5 dB and a rejection of 20 dB at 30 GHz. The main goal of this project is to increase the gap height g to 0.6 mm seen in Fig. 9 to ease manufacturing.

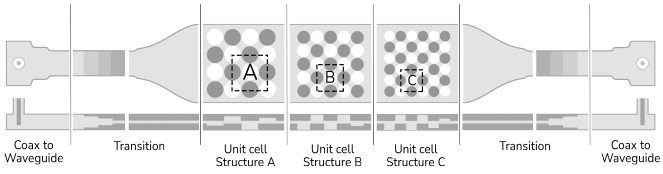


Fig. 8: Schematic overview of the whole filter system, showing all subparts: Coax to waveguide, transition and all three unit cell structures.

A. Unit cell structure

The filtering structure is constructed out of holey glide-symmetric EBG structures, this unit cell structure is of interest due to the easy manufacturing process and its ability to create wide stopbands and high rejection. These EBG structures' electromagnetic properties are determined by their unit cells as previously explained. By altering the different dimensions of the unit cell, such as the hole radius and the periodicity, its dispersive properties will change. Thus making it a tool to introduce desired stopbands. The attenuation and the stopband are however heavily dependent upon the gap height as seen in Fig. 9 and it is therefore a challenge to increase it.

Due to the anisotropic properties of the holey glide-symmetric unit cell different stopbands will be achieved depending on which direction the wave is propagating inside the

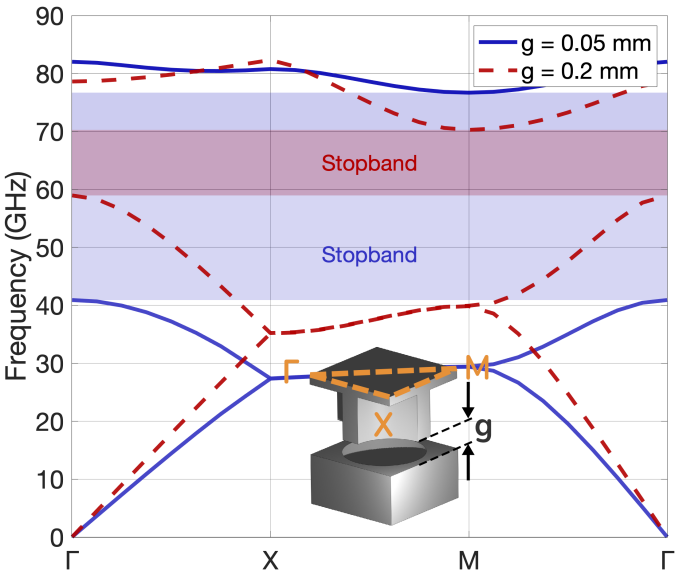
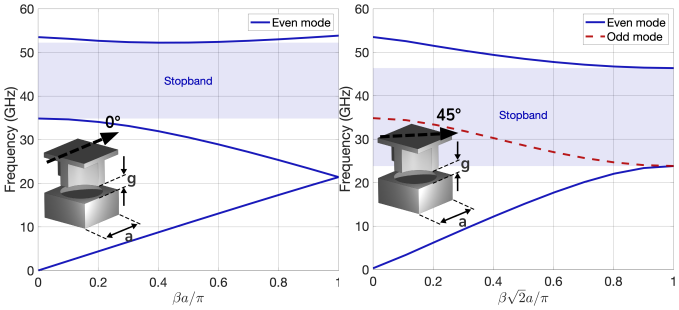


Fig. 9: Dispersion diagram of unit cell with parameters $a = 3.7$, $r = 1.7$, $h = 2$ for $g = 0.05$ mm and $g = 0.2$ mm plotted in blue and red color respectively, simulated in CST Microwave studio eigenmode solver.

structure. As shown in [5] it can be concluded that propagation in the Γ to X (45 degrees) direction results in a wider stopband. This because of the new mode that is introduced as seen in Fig. 10, this mode can be allowed to propagate since it is an odd mode which can be seen in 5 together with the even modes. The odd modes can be allowed to propagate since they can not be picked up or excited by the coaxial cable.



(a) (b)

Fig. 10: a) displaying the dispersion diagram of the conventional unit cell in the 0 degrees direction (Γ to X direction), with a gap height g of 0.2 mm. b) displaying the dispersion diagram of the same unit cell in a) but in the 45 degrees direction (Γ to M direction), simulated in CST Microwave studio eigenmode solver.

Only allowing wave propagation in the 45 degrees direction is therefore the optimal choice if the gap height g is to be increased. Thus another unit cell can be implemented to achieve the same results but in the 0 degrees direction, the so called 45 degrees unit cell. The 45 degrees unit cell can be derived from the already existing unit cell structure as can be seen in Fig. 11.

This is however not enough to create a stopband from 29-60 GHz. To further increase it, three independent unit cell structures A, B and C has to be designed that together can

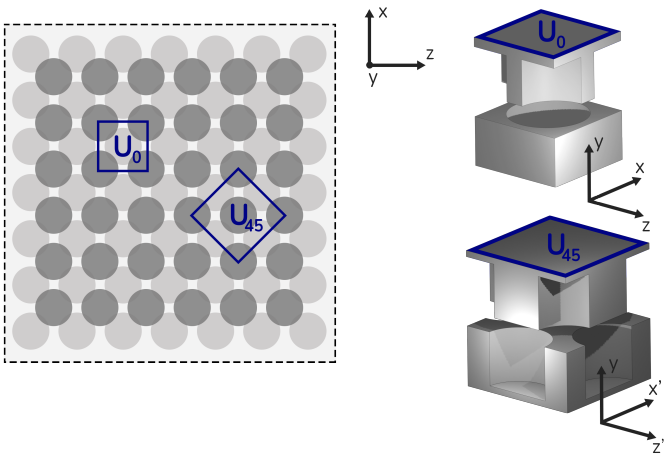
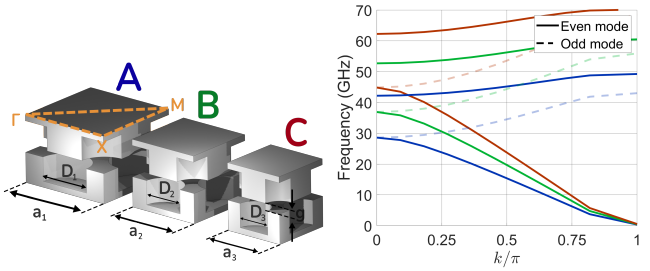


Fig. 11: Glide-symmetric holey EBG structure to the left and the two unit cells to the right: 0 degrees unit cell and 45 degrees unit cell respectively.

cover the whole stopband as seen in Fig. 12.



(a) (b)

Fig. 12: $k = \beta_A a_1 = \beta_B a_2 = \beta_C a_3$ a) displaying the three unit cell designs used in the filtering structure with the following parameters in mm $a_1 = 7.92$, $a_2 = 6.22$, $a_3 = 5.23$, $D_1 = 4.4$, $D_2 = 3.4$, $D_3 = 2.8$, $g = 0.6$. b) displaying the dispersion diagram of the three unit cells in the M to X direction simulated in CST Microwave studio eigenmode solver.

In order to have high attenuation in the beginning of the stopband the unit cell structure covering that part of the stopband, A, had to be longer since it results in a higher rejection. The lengths of the two other structures, B and C, are decided through extensive parametric studies. However, due to impedance mismatches in the filter there will be unwanted reflections in the passband region. Four matching networks are therefore introduced in order to get good transmission in the passband. These matching networks are designed by gradually changing the hole depth h and by doing so changing the impedance. The hole depths is decided by studying the dispersion diagrams for the unit cell with different hole depths. The project group concluded that small phase shifts in the passband region for two different hole depths result in a good match. The resulting S_{21} parameter for the matched filter can be seen in Fig. 13 where the filter fulfills all the requirements with an insertion loss of 0.3 dB and a rejection of 20 dB at 30 GHz

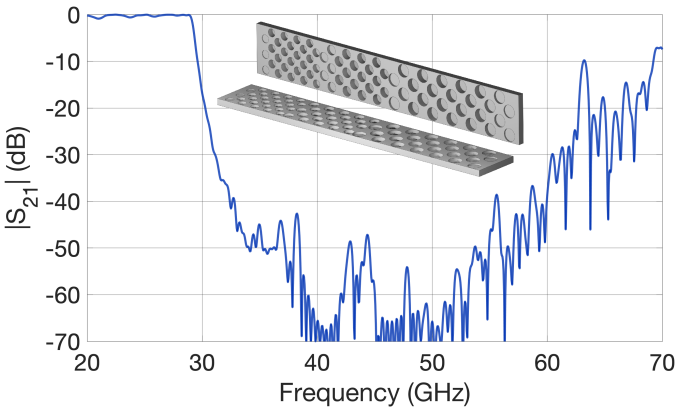


Fig. 13: S_{21} for matched unit cell structure, simulated in CST Microwave studio time domain solver.

B. *Coaxial cable to waveguide transition*

Connecting the filter to standard connections can be done in many ways for example with a waveguide to waveguide adapter. In this report a coaxial connector to waveguide transition is considered and developed. This means that the filter can easily be measured with a vector network analyzer (VNA) since most VNA's use coaxial connectors.

To connect coaxial cables to the filter, an already defined and constructed part is used. The part provides an SMA input and a pin to be placed in a cylindrical waveguide. The pin from the part, together with the cylindrical waveguide, creates a coaxial transmission line. The part and the schematic can be seen in [11]. However, this part can not work on its own. Instead a transition between the connector and the waveguide has to be implemented.

The coaxial connector to waveguide transition consists of two parts: One coaxial transmission line and a rectangular waveguide structure designed to match the impedance of the coaxial transmission line, as can be seen in Fig. 14.

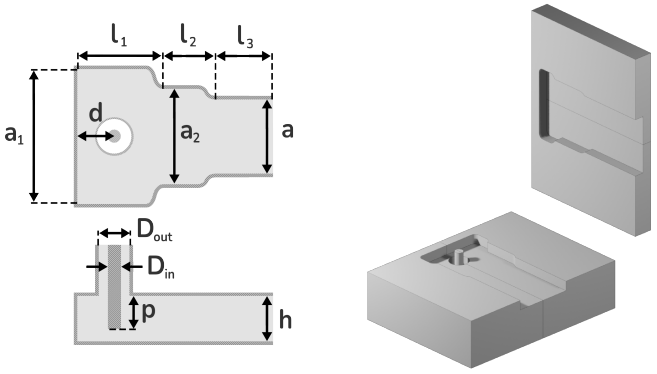


Fig. 14: Schematic of the coax to waveguide transition to the left and 3D model to the right.

First of all the coaxial transmission line's outer diameter is determined with respect to the impedance equation of a coaxial transmission line, Equation 10, as can be seen in II-G. Since the diameter of the inner conductor of the coaxial waveguide (D_{in}) is already determined and the characteristic

impedance is set to 50 Ω, the outer diameter can be calculated to approximately 2 mm.

The next part to design is the waveguide structure. To ensure low reflections in the structure the waveguide structure needs to match the impedance of the coaxial transmission line. An impedance match can be acquired when the propagation constant of the traveling wave is not changing drastically. This impedance matching is done partially by connecting the coaxial transmission line with a waveguide with a larger width than desired. By studying the waveguide dispersion diagram in II-E it can be concluded that the dispersion of TE_{10} is more like that one of a TEM mode in frequencies much higher than the cutoff frequency. In this case the starting width of the waveguide is 7.4 mm, which results in a cutoff frequency much lower than desired. The width of the waveguide is then shortened in two steps to the desired width of 5.95 mm.

The pin that excites the propagating mode in the waveguide structure has to be placed in such a way that high transmission can be made with very low losses. Furthermore due to the positioning and geometry of the pin in the waveguide as seen in Fig. 14 only even modes will be able to be picked up or excited. Another important factor is the distance d from the pin to the metallic wall or backshort since constructive interference with the waves that reflect off this wall is desired. Upon optimizing these designs in CST Microwave studio's time domain solver, a fully working coaxial to waveguide transition can be assured with a maximum insertion loss of 0.2 dB as displayed in Fig. 15.

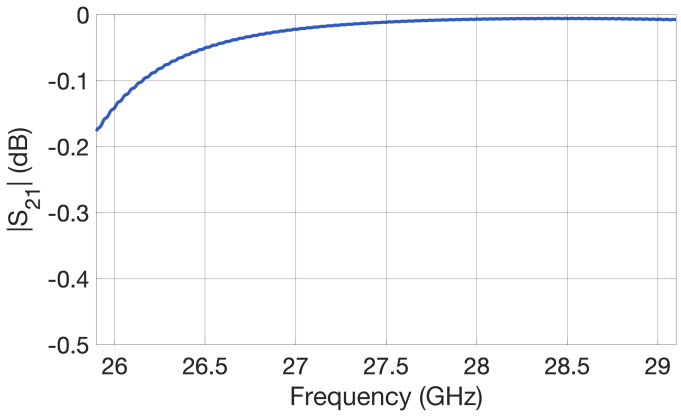


Fig. 15: S_{21} of the coax to waveguide transition, simulated in CST Microwave studio time domain solver, with parameters d = 1.7mm, l_1 = 2.2, l_2 = 4.1 mm, l_3 = 7 mm, a_1 = 7.4 mm, a_2 = 6.6 mm, a_3 = 5.95 mm, D_{out} = 2 mm, D_{in} = 0.86 mm, h = 2 mm and p = 1.55 mm.

C. Stepped impedance transformer and tapering structure

Due to the difference in height between the waveguide and filtering structure a structure matching the heights as well as impedance is needed. To ease manufacturing a stepped impedance transformer can be implemented to solve this kind of problem. A stepped impedance transformer is a structure that changes the impedance gradually by creating a stairlike appearance as seen in Fig. 16.

This structure also has the width of 5.95 mm in order to act as a highpass filter for the lower cutoff at 26 GHz. The height

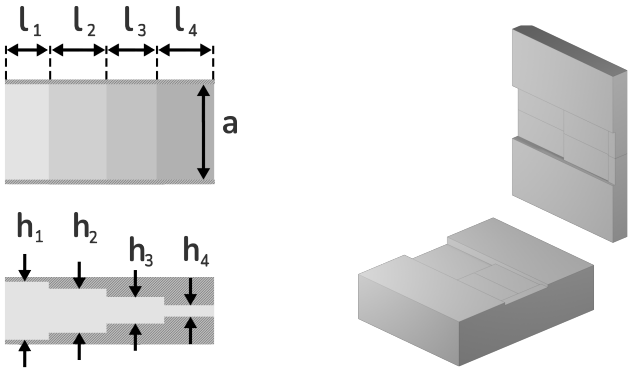


Fig. 16: Schematic over the stepped transformer to the left and the 3D model to the right.

has to be changed from $h = 2$ mm to $h = 0.6$ mm where the number of steps is decided, by carrying out extensive parametric studies, to four steps. After an extensive parametric study has been carried out, the values for the different heights and lengths are: l_1 = 1 mm, l_2 = 7.1 mm, l_3 = 6.5 mm, l_4 = 1 mm, h_1 = 2 mm, h_2 = 1.6 mm, h_3 = 0.9 mm and h_4 = 0.6 mm. This results in a stepped transition with a maximum insertion loss of 0.4 dB, as displayed in Fig. 17 a).

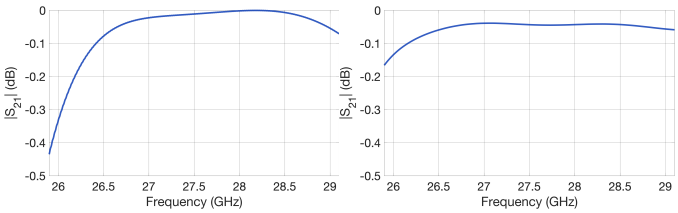


Fig. 17: a) displaying the S_{21} of the stepped transformer with parameters h_1 = 2 mm, h_2 = 1.6 mm, h_3 = 0.9 mm, h_4 = 0.6 mm, l_1 = 1 mm, l_2 = 7.1 mm, l_3 = 6.5 mm and l_4 = 1 mm and b) displaying the S_{21} of the tapering structure with parameters l_1 = 3 mm, l_2 = 1.75 mm, l_3 = 3 mm , l_4 = 20 mm, l_5 = 6 mm, a_1 = 5.95 mm, a_2 = 6.3 mm, a_3 = 16.7 mm and g = 0.6 mm, both simulated in CST Microwave studio time domain solver.

To connect the stepped transformer to the filtering structure a taper has to be implemented in between because of the difference in width. The proposed taper has a width change from 5.95 mm (the width of the stepped transformer) to 16.7 mm (the width of the filter). The proposed taper increases the width slowly at first with a middle width, a_2 of 6.3 mm as seen in Fig. 18. Then the taper increases the width more drastically but still over a longer length to prevent reflections. The implemented taper has a maximum insertion loss of 0.2 dB as displayed in Fig. 17 b).

D. Full system simulation

When combining the transition with the unit cell structure we get the proper passband from 26-29 GHz and a stopband from 29-60 GHz. The filter in its entirety can be seen in Fig. 19 together with the scattering parameters for it. It has an insertion loss of 0.5 dB and a rejection of 20 at 30 GHz. The

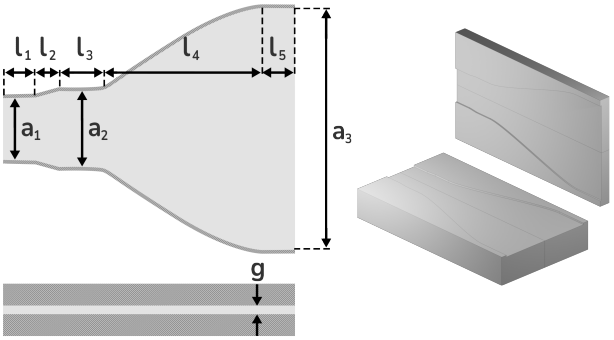


Fig. 18: Schematic over the tapering structure to the left and the 3D model to the right.

filter attenuation in the stopband is however less, with spikes reaching as high as -25 dB.

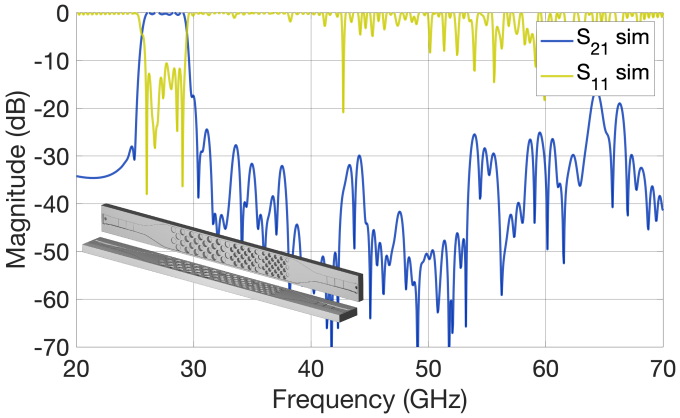


Fig. 19: Simulated scattering parameters for the whole system using CST Microwave studio time domain solver, with 3D render of the final design.

IV. Discussion

A. Design implication

This work shows that a seemingly simple change to an already existing and examined unit cell has a great impact on the performance, as being concluded in [5]. By using multiple holey glide-symmetric 45 degrees rotated unit cell designs, it is possible to increase the crucial air gap between the two structures making up the filter structure. A glide-symmetric holey EBG-filter using one conventional 0 degrees unit cell design require a gap height of 0.05 mm to provide the same stopband as the design that has been provided in this work. In this case, the air gap was extended from 0.05 mm to 0.6 mm. Furthermore, a gap height of 0.6 mm is not possible to achieve with the conventional 0 degrees unit cell.

B. Interpretation of simulation results

When the filter in its entirety was simulated the project group noticed how the filter's rejection in the stopband was lesser, with peaks reaching as high as -25 dB. One possible theory to explain this was that new resonances were introduced when combining the filter with the coax transition. This is however something that would not show up in reality where no perfect electric conductors exist. Another explanation of these peaks is that complex modes which are not visible in the dispersion diagrams produced by CST Eigenmode solver are propagating in the structure and are not affected by the filtering structure.

Analyzing the energy in the system after a simulation shows that some energy gets trapped in the structure. This results in an uncertainty within the results since it remains unknown where that energy would go giving it enough time. In the case of the simulation of the whole system only 8 percent of the energy is unaccounted for. This still gives a result fulfilling the requirements, but the result is only certain to -20 dB.

C. Industry applications

Electromagnetic band gap (EBG) technology is as of now very attractive because of the advantages it brings. With Electromagnetic band gap technology it is possible to implement smart systems broadcasting on higher frequencies without having significant losses. Electromagnetic band gap technology also facilitates the manufacturing process due to the high tolerances in gaps, compared to non-electromagnetic band gap structures. Glide symmetric holey EBG facilitates the manufacturing process even more due to the lack of small pins.

D. Future work

The filter proposed in this report has a large form factor and further studies and new ways of making the structure more compact is needed. Further studies regarding the unit cells, exploring new symmetries and configuration of holes could lead to even better performance regarding filtering with even wider stopbands and higher attenuation. To reduce leakage on the sides of the filter a new unit cell could be designed to create a stopband in the passband region and prevent those frequencies from leaking out of the filter. Although the main goal of this project is to increase the gap height between the two plates and this is done further studies into how to increase it even more is needed.

V. Conclusion

It has been proven that the multiple unit cell design method can be implemented for creating wide stopbands and by doing so also allow the gap height to be increased. The project group concluded that the attenuation was heavily dependent upon the gap height. Because of this filtering structure's length needs to be increased in order to achieve the same attenuation. It has also been concluded that using the 45 degrees configuration is desired if possible because of its ability to create wider stopbands and higher attenuation.

Acknowledgment

The authors would like to thank their supervisors Pilar Castillo Tapia and Freysteinn Vidar Vidarsson for helping out and being supportive during the project. The authors would also like to thank Max Eichenberger and Mathias Axelsson for helping us with the manfuacturing stage.

REFERENCES

[1] Y. Wang, J. Li, L. Huang, Y. Jing, A. Georgakopoulos, and P. Demestichas, "5G mobile: Spectrum broadening to higher-frequency bands to support high data rates," *IEEE Vehicular Technology Magazine*, vol. 9, no. 3, pp. 39–46, 2014.

[2] A. Monje-Real, N. J. G. Fonseca, O. Zetterstrom, E. Pucci, and O. Quevedo-Teruel, "Holey glide-symmetric filters for 5G at millimeter-wave frequencies," *IEEE Microwave and Wireless Components Letters*, vol. 30, no. 1, pp. 31–34, 2020.

[3] E. D. Sharp, "A high-power wide-band waffle-iron filter," *IEEE Transactions on Microwave Theory and Techniques*, vol. 11, no. 2, pp. 111–116, 1963.

[4] O. Quevedo-Teruel, "Periodic structures with glide symmetry and their application to antenna design," in *2020 International Workshop on Antenna Technology (iWAT)*, Bucharest, Romania, 2020, pp. 1–4.

[5] Q. Chen, F. Mesa, X. Yin, and O. Quevedo-Teruel, "Accurate characterization and design guidelines of glide-symmetric holey EBG," *IEEE Transactions on Microwave Theory and Techniques*, vol. 68, no. 12, pp. 4984–4994, 2020.

[6] M. Ebrahimpouri, O. Quevedo-Teruel, and E. Rajo-Iglesias, "Design guidelines for gap waveguide technology based on glide-symmetric holey structures," *IEEE Microwave and Wireless Components Letters*, vol. 27, no. 6, pp. 542–544, 2017.

[7] M. Ebrahimpouri, A. Algaba Brazalez, L. Manholm, and O. Quevedo-Teruel, "Using glide-symmetric holes to reduce leakage between waveguide flanges," *IEEE Microwave and Wireless Components Letters*, vol. 28, no. 6, pp. 473–475, 2018.

[8] Q. Chen, O. Zetterstrom, E. Pucci, A. Palomares-Caballero, P. Padilla, and O. Quevedo-Teruel, "Glide-symmetric holey leaky-wave antenna with low dispersion for 60 ghz point-to-point communications," *IEEE Transactions on Antennas and Propagation*, vol. 68, no. 3, pp. 1925–1936, 2020.

[9] M. Ebrahimpouri, E. Rajo-Iglesias, Z. Sipus, and O. Quevedo-Teruel, "Cost-effective gap waveguide technology based on glide-symmetric holey EBG structures," *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, no. 2, pp. 927–934, 2018.

[10] D. Cheng, *Field and Wave Electromagnetics*, 2nd ed., ser. Addison-Wesley series in electrical engineering. Boston: Addison-Wesley, 1989, pp. 547–554. [Online]. Available: https://books.google.se/books?id=6xmoMgEACAAJ

[11] *2.4mm Straight Female, Air Line Through the Wall, 4 Hole Flange Step-Down Regulator*, Cmpter Electronics Co., Ltd, 2016. [Online]. Available: https://www.cmpter.com/uploadfile/2019/0723/20190723102837354.pdf

CONTEXT J – PART II
DESIGN AND TESTING OF NOVEL MICROWAVE & ANTENNA TECHNOLOGIES

POPULAR DESCRIPTION

Beaming with innovation: how 5G will change the world

With the introduction of a new generation of mobile communication, visions of our future are soon a reality. Smart cities, self-driving cars, and efficient industries reveal only the tip of the iceberg of opportunities.

Wireless communication is already an integral part of society with cell phones being one of the most impressive uses. Two people on the opposite side of the planet communicating in real time is something people in the early 20th century could not even dream about. In a future where vehicles communicate with both each other and infrastructure, wireless communication is fundamental.

The everyday 5G user will enjoy luxuries such as higher quality, low latency video calls and maybe even having such a call through your smart fridge. In addition, for safety critical applications such as self-driving cars, low latency and reliability are imperative and existing solutions simply do not cut it.

Higher transfer rates entails communicating at higher frequencies. This brings about the problem that higher frequency waves are more easily obstructed requiring improved antennas that can better direct its focus toward the receiver. So what types of antenna designs meet these criteria, and are they safe? This is why part of the project includes a study of how radio waves utilized by 5G affect the human body.

To make an antenna that can be used to build these new innovations, we have worked with a type of antenna called leaky wave antennas. One can think of a leaky wave antenna as a water hose leaking water through small holes cut along its length, where instead of water a wireless signal is sent. By sending waves with different frequencies into the antenna, the direction of the signal will change. Leaky wave antennas are easy to build, and can send their signal in specific directions, concluding that leaky wave antennas are an effective design to meet the criteria.

Evidently, 5G will bring a variety of technological advances. It will improve the services we use today, as well as make entirely new technologies possible. With a speed many times higher than the one you have today, the sky is the limit for your future internet use.

SUMMARY OF PROJECT RESULTS

The new generation of mobile communications, 5G, was first deployed at the end of 2019. However, this first implementation will not be able to cover all of the user needs. Further research to increase the capacity of wireless communications systems is currently needed to cover the demand of the near future.

These developments in telecommunication technology have led to a rising demand for antenna designs that can fulfill the new requirements in bandwidth, gain and size. Contemporary methods for achieving a focused beam utilizes dipole antenna arrays, but leaky wave antenna designs have shown great potential for widespread use in new antenna applications, such as automotive radars and 5G, including Internet of Things. The focus of project groups J3a and J3b has been to design two different types of leaky wave antennas.

The increased demand for wireless communication capacity accelerates the deployment of 5G technology, but before this can be done the safety for humans has to be scientifically assured. The method of safety testing includes manufacturing a material with EM properties of human skin and then measuring the energy absorption in the material, which is what the project groups J4 have done.

The project group J3a has designed a linear one dimensional leaky wave antenna, implemented through a waveguide opened on one side, with a tapered row of pins to adjust the leakage. The group worked to achieve antenna parameters that were in accordance with those given by a theoretical model of a leaky wave antenna, such as directivity, side lobe levels, and radiation efficiency.

In the project J3b the group aims to produce a structure achieving broadside and pencil beam radiation, using a two dimensional leaky wave structure based on continuous transverse stubs. As the main use of the design is for establishing communication channels, the design priorities are to achieve a frequency independent beam with as large bandwidth as possible. The design utilizes a corrugated parallel plate waveguide to generate a propagating slow wave. Coupling stub elements are periodically spaced one wavelength apart to provide in-phase excitation and the height of the coupling stub elements are tapered to produce a uniform power distribution, together creating the desired radiation pattern.

The two antennas designed by the J3 groups can be combined, using the antenna of group J3a as a feeding structure for the antenna designed by project group J3b. The combined antenna will thus be able to be fed by a single standard waveguide, and provide broadside radiation capabilities resulting in a highly directive antenna and a structure that is suitable for mass production.

To further research the subject, the design made by the J3b group could be altered to utilize a bed of nails structure instead of corrugations to allow a similar structure to the one produced by the J3a group to control the J3b structure and produce sweeping capabilities.

For the linear antenna examined by project group J3a, leakage radiation in an unwanted direction is commonly suppressed with an electromagnetic band gap structure, which consists of rows of periodic holes on the opposite side of the waveguide opening. Another possible improvement is modifying the leakage tapering structure, in order to use a tapered row of holes in the waveguide wall instead of pins. This structure is less fragile and may be easier to manufacture, especially at scale.

The groups in project J4 have manufactured a phantom, which is to emulate the electromagnetic properties of human skin. A phantom is a dummy sample that serves to be radiated in place of actual human beings. New highly directive 5G antennas are to be employed for industrial as well as commercial use. The specific absorption rate (SAR) of the phantom was measured in an anechoic chamber to determine a potential health risk factor. The electromagnetic properties and SAR were determined by radiating the phantom with a directive antenna and comparing the resulting temperature rise on the surface of the phantom due to the high frequency radiation to a theoretical model. To verify the aforementioned lab results, extensive computer simulations were done to replicate the results.

The manufactured phantom consists of a mixture of water and agar, but due to limitations in the theoretical models the phantom consists mostly of water. A future improvement here is to extend the theoretical model to better fit the human skin.

A relatively small lab space forced us into a small scale simulation. This is not representative of real world use of 5G antennas, which will not come close to the high values of absorption that we measured. Another future extension of the project would therefore be to more closely resemble the conditions we would find in the real world, which among other things means putting a greater distance between the antenna and the phantom.

These results bring us closer to the end goal of the context, which is to aid in the deployment of these new means of wireless communication that can cover future users' needs.

IMPACT ON SOCIETY AND ENVIRONMENT

The constant rise of new and improved generations of mobile communication requires effective and optimized antennas. As fast connection channels become not only available but also more reliable over larger distances, the physical and economic challenges of establishing high speed access to remote locations are overcome. For individuals in these remote locations improved access can bring an overall improvement in quality of life, gaining both the possibility to work remotely and a reliable access to online services such as banks and health care. The new antennas pave the way for the future of technology such as self-driving cars, Internet of Things, artificial intelligence and more.

A prerequisite for self-driving cars is detection of nearby objects. By implementing radars and highly directive antennas, a more precise detection of nearby objects becomes possible. Furthermore, cars communicating with one another on a larger scale through 5G networks can potentially reduce traffic congestion, which has a positive environmental impact and also increase safety of cars by acquiring a more extensive traffic report in real time. Implementation and usage of 5G technologies as it pertains to higher transfer rates and lower latencies will open up new applications for companies such as remote control devices, making it safer for individuals in certain work related situations. Moreover, development of efficient antennas enable applications in the medical field through non-invasive imaging which have a positive impact on medical treatments.

Antennas are often coupled with electronic devices, which are notable for having large environmental impacts. From mine workers extracting metals in unsafe ways which causes local pollution to greenhouse gas emissions and pollutants released during manufacturing, the list of implications goes on all the way to the scrapping where it might end up in landfills contributing to the ever present problem of e-waste.

In addition to more and better antennas, 5G also requires a denser placement of devices. Unfortunately, this opens up the possibility for a more precise tracking of connected individuals if the technology gets into the wrong hands. By using surveillance technologies, a government or company can spy on its own as well as other countries' citizens. In the up and coming world of IoT-devices this is a major concern, which is why the involved parties must ensure that their technology remains in good hands.

One of the largest investors in improved antenna technologies is the military. The question here is not *if* the technology will be used for military applications but rather *how* it will be used. Improved antenna designs are a requirement for developing autonomous weapon systems and armed robots, which are questionable from both an ethical and moral standpoint.

There is also a somewhat widespread worry regarding the safety of higher frequency wireless communication. To ensure that new wireless technologies are safe, they need to be verified by testing how much radiation they give off and in turn how much of that is absorbed by humans. However, a conflict of interests might arise here, where the companies who develop antenna solutions have an incentive not to report if they are emitting harmful levels of radiation. Even though no evidence of the new antennas being harmful has been found, the issue of their safety is still one that has to be continuously verified by independent parties.

Fully Metallic One Dimensional Uniform Tapered-Pin Leaky-Wave Antenna at 30 GHz

Puya Faghi and Henrik Åkerberg

***Abstract*—This paper describes the design of a one dimensional leaky-wave antenna, with an operating frequency at 30 GHz.**

The antenna consists a waveguide with one of the walls replaced by a semi-open row of pins, allowing power to leak out. The waveguide width and the height of the pins are tapered along the waveguide length, in order to control the antenna's radiation parameters.

The antenna has been modeled and tested, using *CST Microwave Studio* and MathWorks *Matlab*. The final antenna design operates at 30 GHz with an efficiency of 90%, side lobe levels of -26.3dB and a beamwidth 6.4°. For other frequencies in the K_a band the angle of maximum radiation varies, giving the antenna scanning capabilities in one dimension.

***Sammanfattning*—Detta dokument beskriver utformningen av en endimensionell läckvågsantenn med en centerfrekvens på 30 GHz.**

Antennen består av en vågledare där en av väggarna ersatts med tappar som gör att effekt kan stråla ut. Vågledarens bredd och tapparnas höjd ändras längs vågledarens längd för att kontrollera antennens strålningsparametrar.

Antennen har modellerats och testats med *CST Microwave Studio* och MathWorks *Matlab*. Den slutgiltiga antenndesignen fungerar vid 30 GHz med en effektivitet på 90%, sidlobsnivåer på -26.3dB och en strålbredd på 6.4°. För andra frekvenser inom K_a-bandet varierar vinkeln för maximal strålning, vilket tillåter antennen att scanna i en dimension.

***Index Terms*—Leaky-wave antenna, 5G, high frequency, one dimensional antenna, microwave, fully metallic.**

Supervisors: Qiao Chen, Osquar Quevedo-Teruel.

TRITA number: TRITA-EECS-EX-2021:172

I. Introduction

The introduction of the fifth generation of mobile communications has opened up a wide array of new applications [1]. The recent commercialization of 5G means that these technological innovations will soon see deployment into our everyday lives. Examples include Internet of Things devices [2], self-driving vehicles [3] and smart cities [4] made possible with distributed computing and data collection. This has the potential to create more efficient societies and improve people's lives.

This era of ever increasing interconnection is constantly demanding more efficient telecommunication solutions. Companies and institutions are continuously trying to make more efficient communication systems, providing higher data rates or cheaper operating costs. As the congestion on the currently used radio band increases, the need to develop higher frequency technologies arises.

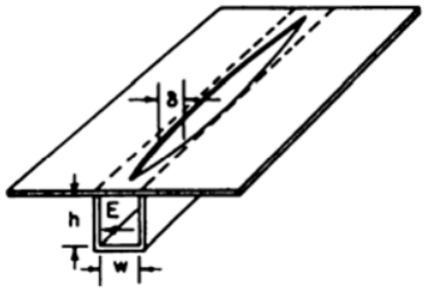


Figure 1. Uniform leaky-wave antenna with tapered slit. Source: [8]

The most common antennas of today use an array of dipole antennas to achieve a focused beam; the beam provides a narrow angular coverage and is steered to a direction by individual phase shifters, one for each dipole [5].

A shift to higher frequencies introduces larger dielectric losses, which in turn makes the aforementioned method inefficient and costly [6]. This introduces the need for new, improved high frequency antennas, preferably ones that use a simple feeding network and have high efficiency for frequencies over 30 GHz [6].

Leaky-wave antennas are a class of antennas that fulfill these requirements and show good potential for use in higher frequency applications. The implementation of a leaky-wave antenna gives a structure that is low profile, simple to feed and provides a high gain at the desired frequencies [7]. The basic operating principle of a leaky-wave antenna is a wave guiding structure, opened in some way in order to radiate power as an electromagnetic wave traverses it. Some dimensions of the structure and its opening can be tapered along its length to modify the leakage rate and the antenna's dispersive properties. This is generally necessary to produce a radiation pattern with low side lobes (undesirable radiation to the sides of the main lobe) [8]. An example of one type of leaky wave antenna is a tapered slit antenna, which is shown (with an exaggerated slit width) in Fig. 1.

Leaky-wave antennas belong to a larger family of antennas collectively referred to as traveling wave antennas. Other antennas in this family include surface-wave and slot array antennas, which share some similarities with leaky-wave antennas, but generally have different design procedures [8]. An earlier article [9] proposes a leaky wave antenna design operating for a point-to-point communication system at mm-wave frequencies.

In this article we will describe the design process of

a leaky wave antenna and compare the simulated antenna characteristics with the theoretically possible ones.

II. Theory

Leaky-wave antennas can be classified into two main types: Uniform or periodic structures. A uniform antenna has a uniform shape along its length, while a periodic antenna has a periodic modulation along its guiding structure [8].

The antenna designed in this paper was set to radiate in one dimension, but leaky wave antennas can also be designed in a two-dimensional radiation pattern [10]. Based on the choice of a uniform one-dimensional radiation pattern for this project, only the characteristics of uniform one-dimensional leaky wave antennas will be discussed further.

A uniform one dimensional leaky-wave antenna operates by first being fed from one end of the antenna. As the wave travels trough the antenna, power radiates through the leaky opening in the side wall [8]. Some of the power does not radiate and reaches the end of the antenna, where it is absorbed by a matched load. The amplitude of the wave therefore decays exponentially, with a leakage rate α and a phase constant β, as given by $A = e^{\alpha + j\beta}$ [8].

The leakage rate α will be adjusted by tapering the open structure of the antenna. Leaky-wave antennas are typically designed with a radiation efficiency of 90%, meaning that 90% of the power is radiated across the antenna's length, and the remaining 10% is absorbed by the matched load [8]. The antenna length is chosen with respect to the radiation efficiency, as well as the range of possible values for α. This usually results in an antenna length of 20 λ_0, where λ_0 is the wavelength of the operating frequency [8]. According to [9], the leakage rate is calculated as shown in (1), using simulated S_{11} and S_{21} parameters, waveguide length L and the free-space wave number (k_0). The S-parameters S_{11} and S_{21} quantify how much of the input power is reflected back to the source and how much of the input power that reaches the end of the antenna, respectively.

$$\frac{\alpha}{k_0} = \frac{-\lambda_0}{L} \frac{\ln(|S_{11}|^2 + |S_{21}|^2)}{4\pi} \tag{1}$$

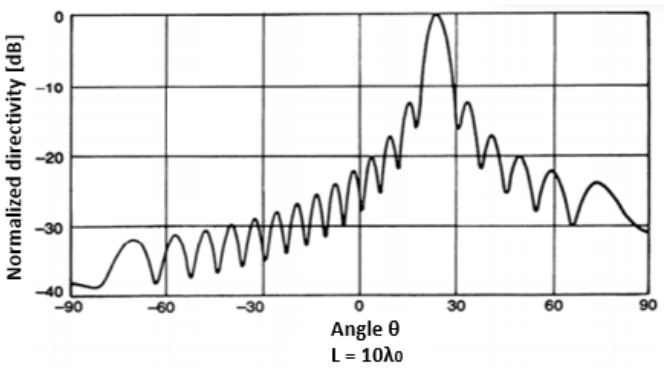
$$k_0 = \frac{2\pi}{\lambda_0} \tag{2}$$

An expression for the antenna radiation pattern (R) is complicated due to side lobe behaviour. It can, however, be shown that as the antenna length goes to infinity the side lobe behaviour disappears [11] (as seen in Fig. 2) and a simpler expression shown in (3) can be observed.

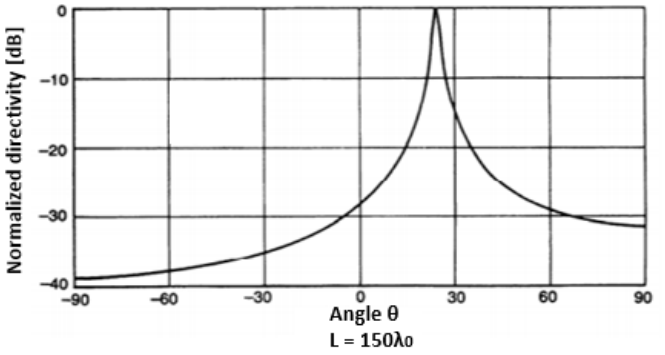
$$\mathrm{R}(\theta) \approx \frac{\cos^2(\theta)}{(\frac{\alpha}{k_0})^2 + (\frac{\beta}{k_0} - \sin(\theta))^2} \tag{3}$$

The antenna is tailored to radiate a beam focused at a maximum radiation direction θ_{max}. According to [8], the angle θ_{max} can be expressed in terms of the phase constant β and k_0:

$$\theta_{max} = \arcsin(\frac{\beta}{k_0}) \tag{4}$$



(a)



(b)

Figure 2. Radiation patterns of dielectric grating leaky-wave antennas, showing the changes in the side lobe behaviour for antenna length a) L = $10\lambda_0$ and b) $L = 150\lambda_0$. Source: [11]

As seen in (2) and (4), the leaky-wave antenna's angle of maximum radiation will scan with a variation in frequency. β at 30 GHz will in turn be gathered from a simulated dispersion diagram. A dispersion diagram is a graphic representation of the relationship of frequency to wavelength, and can be simulated for a specified antenna geometry.

Since a defining property of a leaky-wave antenna is that the radiation is generated by a fast wave ($\beta < k_0$), the leaky-wave antenna can be treated as a linear array with very small element spacings [9]. Therefore, the leakage rate α can be expressed as in [8] in terms of the aperture illumination magnitude M(z), the desired radiation efficiency η and the position along the waveguide length z as seen in (6).

The aperture illumination M(z) is a measure of how much power is radiated at each length z on the antenna. To make optimal use of the antenna's aperture and to reduce side lobes, the aperture illumination M(z) should follow a sinusoidal pattern (5) as in [12]. Since the power decays along the antenna's length, the leakage rate needs to be tailored accordingly to achieve a correct illumination distribution. An illumination as in (5) results in α values that are plotted along with the illumination M(z) in Fig. 3. Since the overall behaviour of the graphs are more important than their particular amplitudes, their amplitudes are normalized in this graph.

$$\mathrm{M}(z) = \sin(\frac{z\pi}{L}) \tag{5}$$

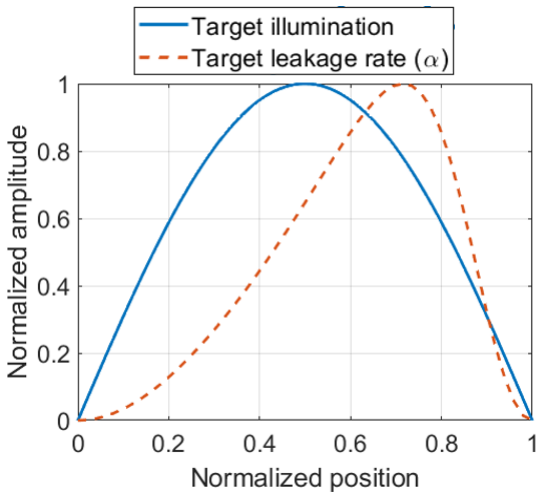


Figure 3. Target illumination and leakage rate along the antenna

$$\alpha(z) = \frac{0.5|\mathrm{M}(z)|^2}{\frac{1}{\eta}\int_0^L |\mathrm{M}(\zeta)|^2 d\zeta - \int_0^z |\mathrm{M}(\zeta)|^2 d\zeta} \tag{6}$$

The radiation pattern's directivity D(z) can be calculated as:

$$\mathrm{D}(z) = \int_0^z \mathrm{M}(\zeta) e^{jk_0 z sin(\theta)} d\zeta \tag{7}$$

III. Parametric Studies

A. Goals

The goal set for the project was to design a one dimensional uniform leaky-wave antenna. In order to achieve the best possible radiation characteristics it was necessary to control the dispersion parameters α and β by tapering the antenna dimensions. The project aim was for a typical radiation efficiency of 90%, with a sinusoidal illumination. With the desired illumination M(z) and a (preliminary) antenna length of 200 mm, (6) gives us a maximum leakage rate of $\alpha = 0.0175$. As for β, the target was to keep the value of β constant through the entire antenna, in order to keep the angle of maximum radiation constant according to (4).

The goal of the design process was to achieve an antenna with performance comparable to the theoretical limits. The antenna was to show side lobe levels of less than -20dB of the main lobe directivity.

B. Modeling and simulation

Two test models were created, a slit antenna (Fig. 4) and a pin antenna (Fig. 5). The tapering of the antenna was originally set to be achieved by having an open slit of varied width, but in order to get a better simultaneous control of α and β, and because the slit model showed irregular behaviour for low values of α, the project group later chose to use a tapered row of square pins as the leaky mechanism of the antenna, as well as tapering the waveguide height. This two dimensional tapering allowed for a more precise tailoring of the radiation pattern, and the pin design allowed for smaller values of α without simulation errors or irregularities.

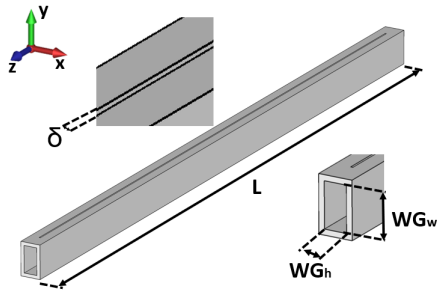


Figure 4. Slit antenna

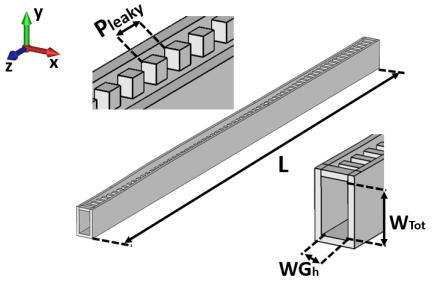


Figure 5. Pin antenna

The slit antenna consists of a waveguide with a uniform slit across the sidewall, enabling power to leak out of the slit depending on the slit width δ. The slit is opened up from $z = 2\,\mathrm{mm}$ to $z = 198\,\mathrm{mm}$, with $2\,\mathrm{mm}$ completely closed off at each end to prevent simulation errors from the excitation ports being too close to the opening of the antenna.

The pin antenna consists of a waveguide with a completely opened sidewall. The open sidewall of the antenna contains a row of uniformly spaced square pins. The leakage rate from the the antenna is in this case controlled by changing the height P_h of the squared pins (Fig. 6). The waveguide is, like the slit antenna, closed off by two lids $2\,\mathrm{mm}$ from each end of the antenna. No pins are situated under the lids, and the waveguide width is therefore slightly larger at the ends: W_{Tot} in Fig. 5 is defined from the lid, which is thinner than a pin ($1\,\mathrm{mm}$ compared to $1.4\,\mathrm{mm}$) but starts at the very top of the structure, while WG_w in Fig. 6 is defined from a pin which is thicker but starts $0.5\,\mathrm{mm}$ from the top of the opening.

To study the dispersion parameters, *CST Microwave Studio* was used, and the data gathered from simulations was processed using MathWorks *Matlab*. α was simulated using

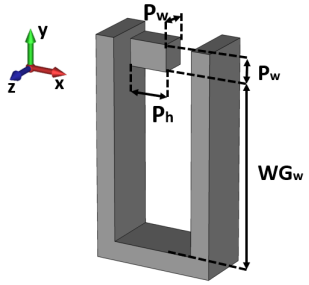


Figure 6. Cross section of pin antenna

Table I
SLIT TEST ANTENNA DIMENSIONS

WG_w	8.0 mm
WG_h	4.0 mm
L	200 mm
Wall thickness	1.0 mm
δ	0.20 mm

Table II
PIN TEST ANTENNA DIMENSIONS

WG_w	7.1 mm
W_{Tot}	8.0 mm
WG_h	4.0 mm
L	200 mm
Wall thickness	1.0 mm
P_h	2.0 mm
P_{leaky}	3.0 mm
P_w	1.4 mm

the *time domain solver*. Since the antenna is to be fed by a waveguide, this was represented in the simulation with waveguide ports at $z = 0\,\mathrm{mm}$ and $z = 200\,\mathrm{mm}$, and the boundary conditions were set to *Open (add space)*, corresponding to a perfectly matched layer with minimal reflections. The β value was primarily simulated using the *eigenmode solver*, which is a solver that finds the characteristic modes of propagation for a certain geometry. The *eigenmode* simulations were done with a unit cell (see Fig. 6) with periodic boundaries in the z direction, and PEC (Perfect Electrical Conductor) boundary conditions on the x and y_{min} boundaries, forcing the E-field to be normal to these boundaries. Since the *Open (add space)* boundary condition is not supported by the *eigenmode solver*, a PMC (Perfect Magnetic Conductor) boundary was used at the y_{max} boundary to approximate a perfectly matched layer, since it forces the E-field to be tangential at the boundary.

In addition to simulating β through the *eigenmode solver*, we also calculated β for some frequencies by using (4) with θ_{max} simulated with the *time domain solver*. This way of calculating β agreed with the β from the *eigenmode solver*, as can be seen in Fig. 7 and Fig. 8. The *eigenmode solver* is not capable of simulating α, but the correspondence between θ_{max} and β nevertheless confirms that the results from both solvers are valid and compatible.

Fig. 9 shows that different slit widths causes only small variations in $\beta \cdot p$, which means that tapering the slit width would still result in a quasi-constant β along the slit antenna.

In order to find suitable dimensions for the antenna, parametric sweeps for α were performed on both test antennas, and the normalized leakage rate at 30 GHz was plotted against the swept parameters. One dimension was swept as the others remained constant as of Table I for the slit antenna and Table II for the pin antenna. The result from this study are shown in Fig. 10 (for the slit antenna) and Fig. 11 (for the pin antenna).

Since we were aiming to keep β constant, rather than designing for a particular value, we only investigated the effects on β from the dimensions to be tapered; slit width δ in the case of the slit antenna, and pin height P_h and waveguide width WG_w in the case of the pin antenna.

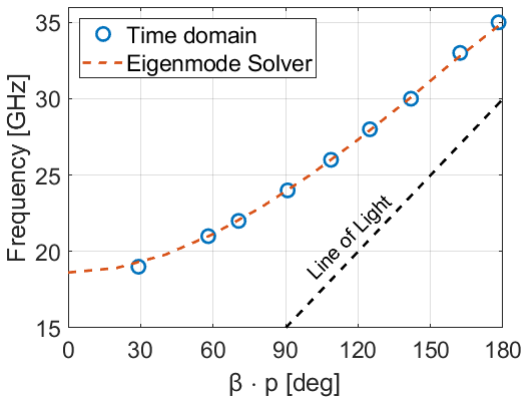


Figure 7. Comparison of $\beta \cdot p$ from eigenmode and time domain solvers for the slit antenna

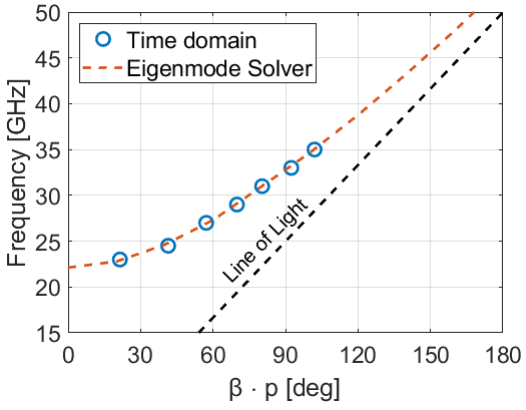


Figure 8. Comparison of $\beta \cdot p$ from eigenmode and time domain solvers for the pin antenna

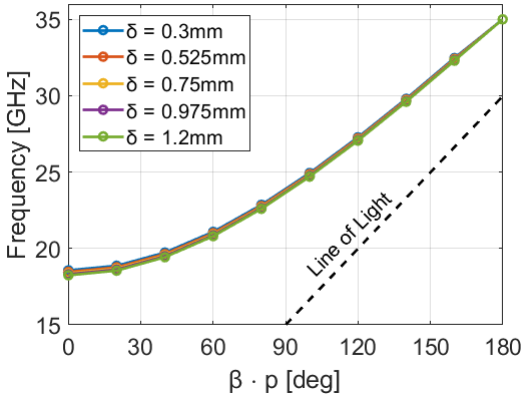


Figure 9. Dispersion diagram of varying slit widths

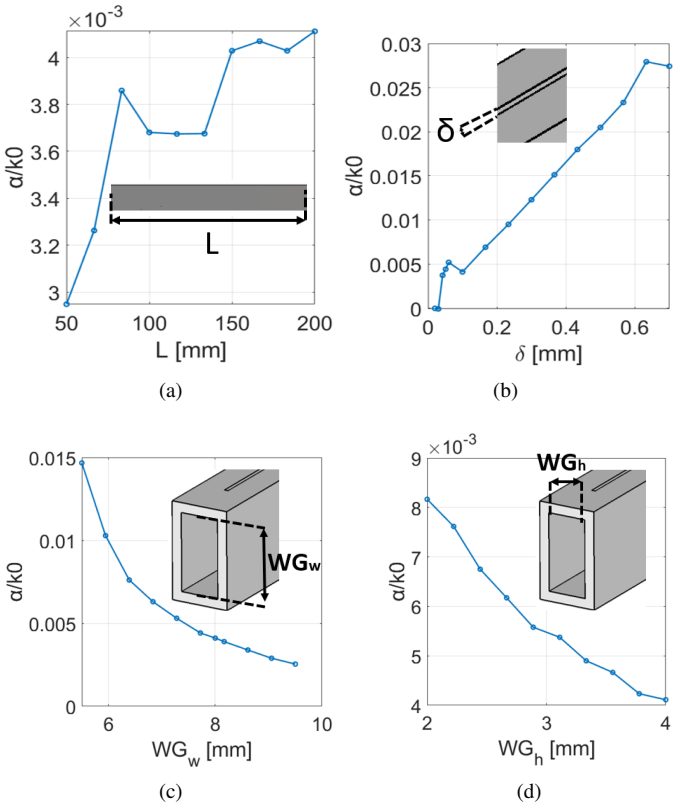


Figure 10. Parametric sweep data for slit test antenna dimensions

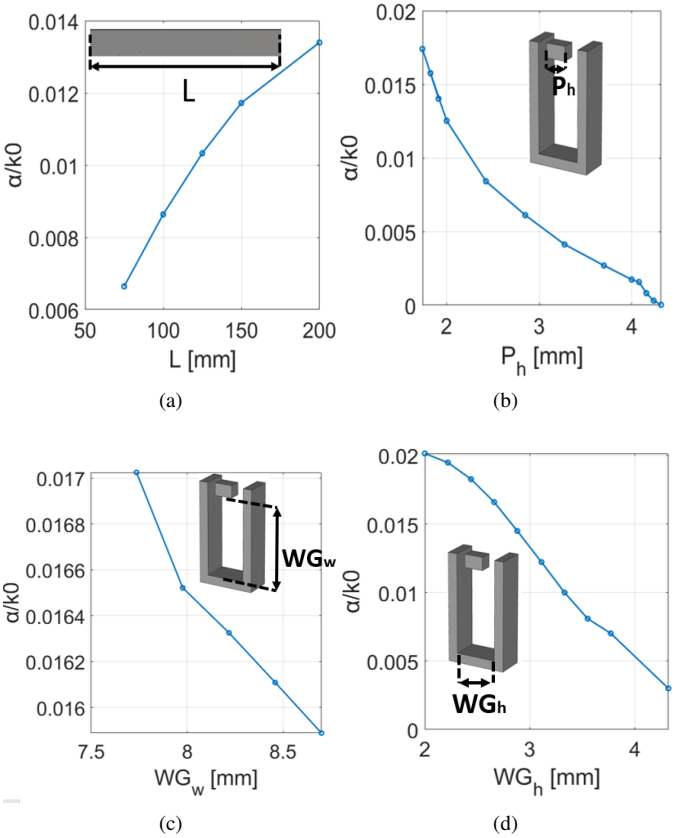


Figure 11. Parametric sweep data for pin test antenna dimensions

Table III
PRELIMINARY ANTENNA DIMENSIONS

WG_w	7.736 mm
WG_h	4.318 mm
L	200 mm
Wall thickness	1.0 mm
P_{leaky}	3.0 mm
P_w	1.4 mm

C. Choice of final antenna dimensions and tapering

From the previous parametric studies (Fig. 10 and Fig. 11) we can observe several patterns in the α characteristics for both antennas.

For the slit antenna, lengths L smaller than 150 mm resulted in irregular α behaviour. Moreover α decreased exponentially with increasing waveguide width WG_w, and decreased linearly with increasing waveguide height WG_h. Finally, α showed a linear correlation with increasing slit width δ, and this correlation gave the largest range of $\frac{\alpha}{k_0}$ values, from 0 to 0.03. For small δ, however, α displayed irregular behaviour. This problem with achieving small leakage rates was likely to cause problems with creating a correct aperture illumination as in Fig. 3. This, paired along with the desire for another degree of freedom in the tapering process, led to us using the pin antenna being considered for the final design.

For the pin antenna there was a fairly linear relation between length L and α, and the irregularities displayed by the slit antenna below 150 mm were not present. α decreased exponentially with increased pin height P_h and it was, unlike the case for the slit antenna, possible to achieve down to $\alpha = 0$ without irregular behaviour. α showed a weak linear decrease with increasing waveguide width WG_w, and decreased relatively linearly with increasing waveguide height WG_h.

A typical length for a leaky-wave antenna is around $20\lambda_0$ [8], which for our operating frequency 30 GHz corresponds to a length $L = 20\lambda_0 = 20 \cdot 10\,\text{mm} = 200\,\text{mm}$. Our α was stable at this point, and this length was also a good tradeoff between antenna performance and practicality/ease of manufacturing. In contrast to the slit antenna, the variation of α through tapering P_h is discrete, which means that a longer antenna that fits more pins can have a smoother tapering profile. This discrete, uneven tapering profile was obviously less desirable than the completely continuous tapering a slit design would have offered, but as noted above only the pin design allowed for the desired range of $\frac{\alpha}{k_0}$ values all the way down to 0.

From the parameter sweeps above we drew the conclusion that a final antenna with dimensions similar to the pin test antenna could, with tapering of P_h and WG_w, achieve the necessary dispersion parameters along its length. The antenna pin width P_w, pin distance P_{leaky} and waveguide height WG_h were therefore kept constant along the antenna length.

In order to use a simple feeding network, the antenna was designed to be fed through a standardised waveguide, and the antenna thus needed to conform to a standardised waveguide size at the ends. The closest standard waveguide size was IEC-R260, with a width of 8.636 mm and a height of 4.318 mm [13]. The antenna height was thus set to 4.318 mm, and the

total antenna width W_{Tot} was set to $8.636\,\mathrm{mm}$, which results in a WG_w of $7.736\,\mathrm{mm}$ ($WG_w = W_{Tot} + \text{thickness} - 0.5\,\mathrm{mm} - P_w$), where the dimension of $0.5\,\mathrm{mm}$ is the distance from the top of the antenna to the top of the pins. From the dimensions mentioned above a preliminary pre-taper antenna is created; its dimensions are shown in Table III.

After the preliminary antenna's dimensions were fixed, the tapering process could begin. As mentioned above, the waveguide width WG_w and pin height P_h were to be varied along the length, in order to achieve the desired radiation characteristics: low side lobes and high directivity. The key to achieving this was to recreate the α profile from Fig. 3 as accurately as possible, and to keep β (and thereby also θ_{max}) as constant as possible.

The tapering is done by first simulating two more sets of data (or lookup tables), one for how α varies by different combinations of values of P_h and WG_w, and a similar data set for β. The data sets will then be interpolated to give continuous curves for α and β, which will then be used to modify the base untapered antenna's dimensions.

The interpolation was done with 4_{th} degree polynomials. The interpolated data sets are plotted in Fig. 12, and the equations for α and β as functions of P_h and WG_w are (8) and (9) respectively. As seen in the graphs and the equations, α is mostly decided by P_h, while WG_w dominates the β function (9). Because of this, the α profile was created through tapering of P_h, and β was kept constant through tapering of WG_w.

While the two different tapered dimensions are dominant for different dispersion parameters, they still have some effect on the other parameter. This led to an approach where a primary P_h and WG_w taper was made, which was then the object of a second tapering step, in order to create a final taper that conformed better to the desired α and β profiles.

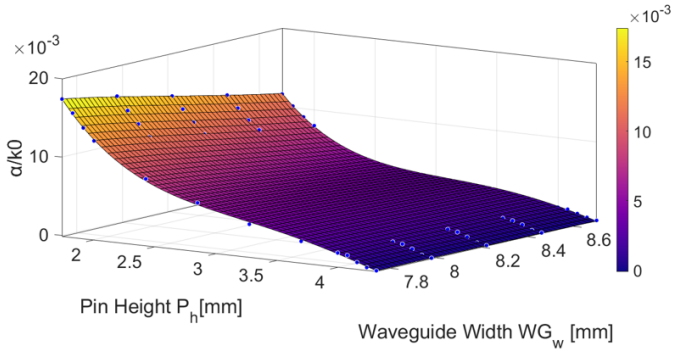
The primary taper profiles were created starting from the untapered antenna's dimensions. Firstly, WG_w was kept constant. By using (6), the tapering profile from Fig. 3 was translated into P_h values that corresponded with the desired α for every point z along the antenna's length. A suitable β was then chosen to be held constant, and the newly created P_h profile was fed into (9) to create a WG_w profile that together with the P_h profile would make up the primary taper.

This process was then repeated, but instead of using the untapered antenna, the primary P_h and WG_w profiles were input into the equations, to create the final taper. At $z \in [0, 2]$ mm and $z \in [198, 200]$ mm on the final taper WG_w was manually forced to conform to the standard waveguide size, that is $WG_w = 7.736\,\mathrm{mm}$. The primary and final tapers are shown in Fig. 13, and a picture of the final model is shown in Fig. 14.

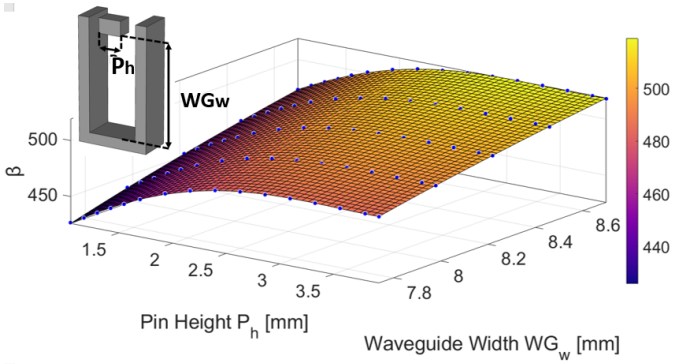
IV. Results

The calculated antenna dimensions (Fig.14) were designed and simulated in the *CST time domain solver*, around the operating frequency of 30 GHz. The simulation results were compared with calculations and are presented here.

The $\frac{\alpha}{k_0}$ leakage profile is shown in Fig. 15a. The final leakage profile clearly follows the target profile with only small



(a)



(b)

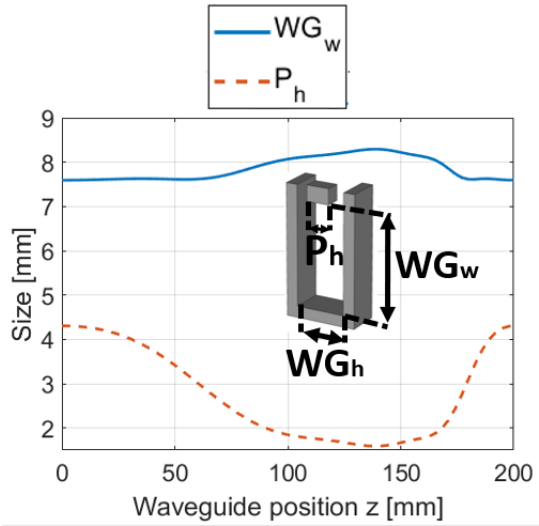
Figure 12. a) Interpolated α surface, b) Interpolated β surface

deviations. The maximum radiation angle θ_{max} is maintained quite constant at $49.9\,\mathrm{deg}$. The close similarity of the final leakage profile creates an illumination that is very similar to the target illumination as seen in Fig. 15b. The antenna therefore has an illumination very close to a sine half period, which creates an optimal aperture utilization, resulting in optimal farfield characteristics.

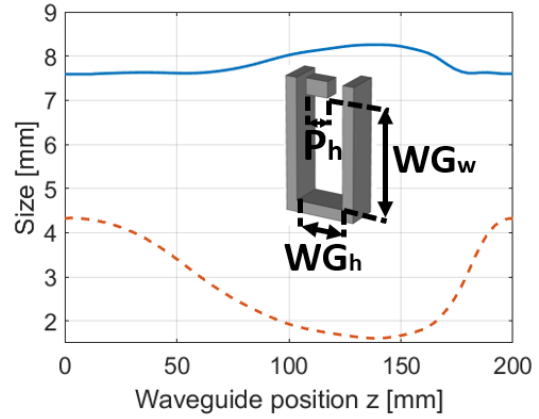
$$\begin{aligned}\alpha(P_h, WG_w) &= -7.6 + 0.25P_h + 3.6WG_w \\ &+ 0.056P_h^2 - 0.15P_hWG_w - 0.62WG_w^2 \\ &- 0.00995P_h^3 + 0.2\cdot10^{-3}P_h^2WG_w \\ &+ 0.019P_hWG_w^2 + 0.047WG_w^3 \\ &+ 6.4\cdot10^{-3}P_h^4 + 30\cdot10^{-6}P_h^3WG_w \\ &- 60\cdot10^{-6}P_h^2WG_w^2 \\ &+ 0.75\cdot10^{-3}P_hWG_w^3 + 1.3\cdot10^{-3}WG_w^4\end{aligned} \tag{8}$$

$$\begin{aligned}\beta(P_h, WG_w) &= -25\cdot10^3 + 310P_h \\ &+ 12\cdot10^3WG_w - 24P_h^2 \\ &- 39P_hWG_w - 2\cdot10^3WG_w^2 - 9.7P_h^3 \\ &+ 6.3P_h^2WG_w + 0.40P_hWG_w^2 \\ &+ 170WG_w^3 + 0.99P_h^4 + 0.31P_h^3WG_w \\ &- 0.36P_h^2WG_w^2 + 0.095P_hWG_w^3 - 5.1WG_w^4\end{aligned} \tag{9}$$

Fig. 16 shows a S_{11} parameter of -27dB and a S_{21} parameter of -10dB at $30\,\mathrm{GHz}$. This shows that the antenna both has a good matching and little reflections, but more importantly that the goal of a 90% radiation efficiency was achieved.



(a)



(b)

Figure 13. a) Primary taper profiles, b) Final taper profiles

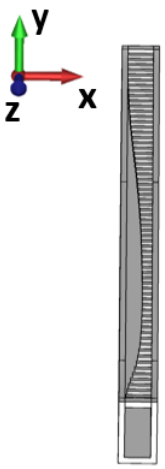
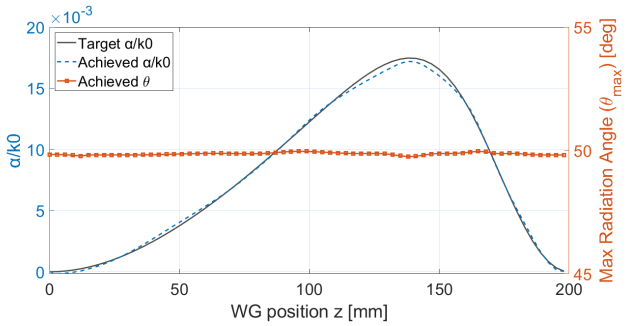
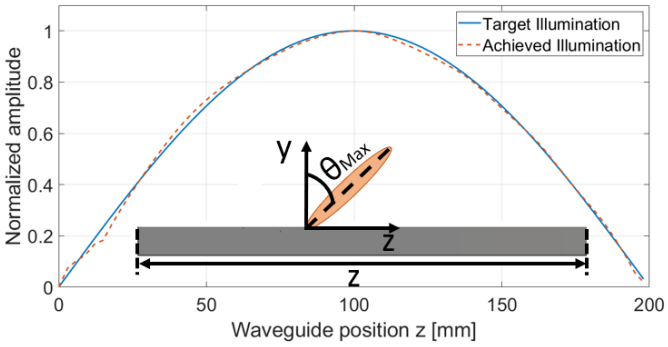


Figure 14. Final antenna structure



(a)



(b)

Figure 15. a) Final α profile compared to target and final θ_{max}, b) Final and target illumination profile

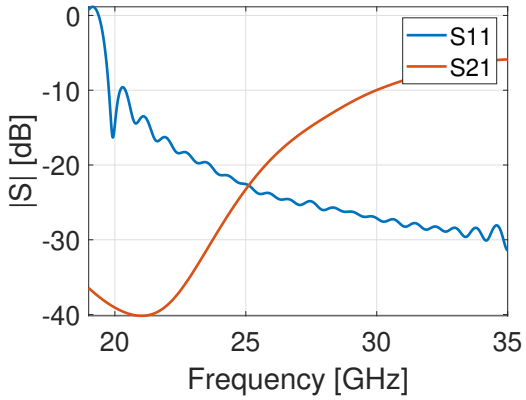


Figure 16. S_{11} and S_{21} parameters

A Comparison between the theoretical farfields and the simulated ones (Fig. 17a and 17b), shows that both have a main lobe direction of $49.9\,\mathrm{deg}$, the same as in Fig. 15a. The theoretical farfield has a beamwidth of 5.4°, while the simulated farfield has a beamwidth of 6.3°. The side lobe levels are both lower than -20dB, achieving the goals set by the project, although the simulated farfield does have a larger beamwidth than the theoretical one. As seen in (4), θ_{max} varies with frequency, and this scanning behaviour is confirmed by the farfield simulations, as can be seen in (Fig. 17c).

V. Discussion

The final antenna design can be used for future high frequency radio and satellite communications as well as 5G

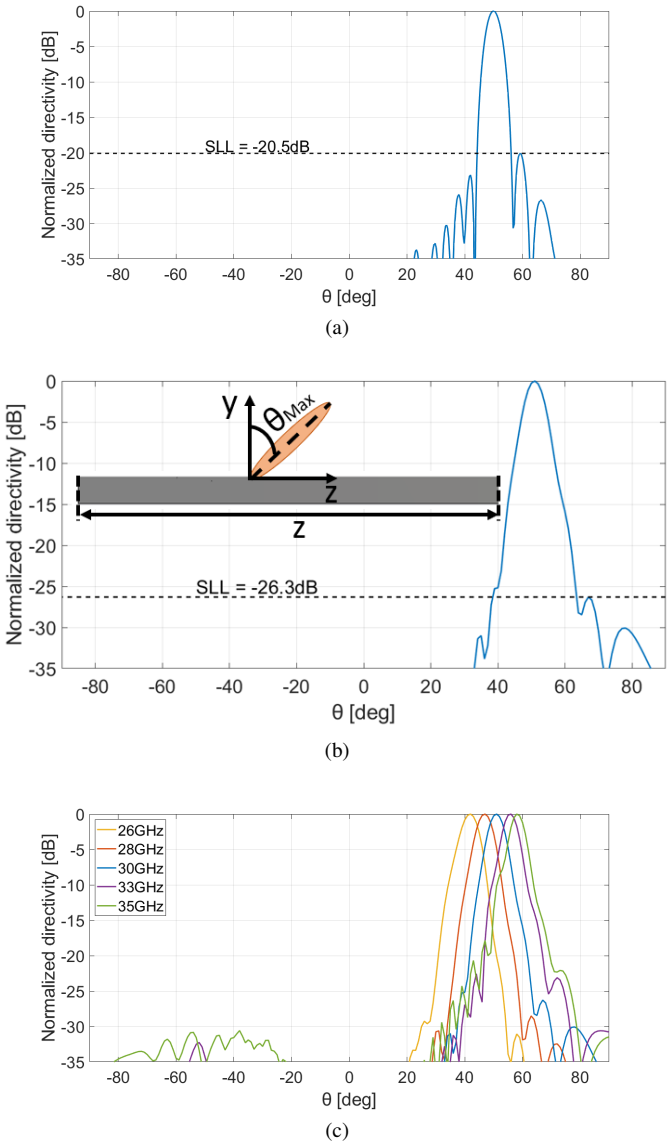


Figure 17. a) Calculated farfield at 30 GHz b) Simulated farfield c) Simulated farfield at different frequencies

and future generational communication systems.

The antenna can be compared with the design of [9]; in this case the antenna from [9] contains a more robust leaky structure, while the pin structure of the antenna proposed in this paper is more fragile, and could be prone to damage during manufacturing or due to ill handling.

A further improvement to the current design is to integrate a dispersive metasurface prism-lens in front of the leaky-wave antenna as in [12] and [9]. The lens would provide for a more focused beam that would lead to a higher gain at the desired radiation angle. According to [14] a gap waveguide is a two-dimensional (2-D) metamaterial structure inside a parallel-plate waveguide which inhibits propagation along the structure. This can be used to create a 2-D structure that will act as a artificial magnetic conductor (AMC) while the parallel plates act as a perfect electric conductor (PEC). If the 2-D structure (in our case, 2-D squared pins) are designed in a periodic manner with the gaps between them being smaller than $\frac{\lambda_0}{4}$ [12], the periodic structure will create a stop band for the electromagnetic waves. This can then be utilized in the side wall without pins to enable manufacturing of the antenna in two parts, while still suppressing undesired backwards radiation.

VI. Conclusion

In this paper, a one dimensional uniform tapered-pin leaky-wave antenna was designed for an operating frequency at 30 GHz. The antenna structure consists of a waveguide where one sidewall is removed and replaced by pins. The smooth tapering profile, together with a tapered waveguide width, provides low side lobes and good directivity. The antenna efficiency is 90%, the beamwidth is 6.4°, and the side lobe levels are -26.3dB.

VII. Acknowledgment

The authors want to thank Qiao Chen for all his guidance and Oscar Quevedo-Teruel for the structure of the context J and his insightful lectures.

References

[1] B. O. hAnnaidh, P. Fitzgerald, H. Berney, R. Lakshmanan, N. Coburn, S. Geary, and B. Mulvey, "Devices and Sensors Applicable to 5G System Implementations," in *2018 IEEE MTT-S International Microwave Workshop Series on 5G Hardware and System Technologies (IMWS-5G)*, Dublin, Ireland, Aug 2018, pp. 1–3.

[2] N. Gupta, S. Sharma, P. K. Juneja, and U. Garg, "SDNFV 5G-IoT: A Framework for the Next Generation 5G enabled IoT," in *2020 International Conference on Advances in Computing, Communication Materials (ICACCM)*, Dehradun, India, Aug 2020, pp. 289–294.

[3] M. Daily, S. Medasani, R. Behringer, and M. Trivedi, "Self-driving cars," *Computer*, vol. 50, no. 12, pp. 18–23, Dec 2017.

[4] R. R. Harmon, E. G. Castro-Leon, and S. Bhide, "Smart cities and the internet of things," in *2015 Portland International Conference on Management of Engineering and Technology (PICMET)*, Portland, USA, Aug 2015, pp. 485–494.

[5] K.-M. Luk and B. Wu, "The magnetoelectric dipole—a wideband antenna for base stations in mobile communications," *Proceedings of the IEEE*, vol. 100, no. 7, pp. 2297–2307, Apr 2012.

[6] O. Quevedo-Teruel, J. Miao, M. Mattsson, A. Algaba-Brazalez, M. Johansson, and L. Manholm, "Glide-Symmetric Fully Metallic Luneburg Lens for 5G Communications at Ka-Band," *IEEE Antennas and Wireless Propagation Letters*, vol. 17, no. 9, pp. 1588–1592, Jul 2018.

[7] D. R. Jackson, P. Baccarelli, P. Burghignoli, A. Galli, and G. Lovat, "Development of leaky-wave antennas," in *2016 IEEE International Symposium on Antennas and Propagation (APSURSI)*, Fajardo, PR, June 2016, pp. 687–688.

[8] D. Jackson and A. Oliner, "Leaky-wave antennas," in *Modern antenna handbook*, C. Balanis, Ed. Hoboken, New Jersey: John Wiley & Sons, Ltd, Nov 2007, pp. 325–367.

[9] Q. Chen, O. Zetterstrom, E. Pucci, A. Palomares-Caballero, P. Padilla, and O. Quevedo-Teruel, "Glide-symmetric holey leaky-wave antenna with low dispersion for 60 ghz point-to-point communications," *IEEE Transactions on Antennas and Propagation*, vol. 68, no. 3, pp. 1925–1936, Oct 2020.

[10] S. K. Podilchak, A. P. Freundorfer, and Y. M. M. Antar, "Broadside radiation from a planar 2-d leaky-wave antenna by practical surface-wave launching," *IEEE Antennas and Wireless Propagation Letters*, vol. 7, pp. 517–520, Oct 2008.

[11] F. K. Schwering and Song-Tsuen Peng, "Design of dielectric grating antennas for millimeter-wave applications," *IEEE Transactions on Microwave Theory and Techniques*, vol. 31, no. 2, pp. 199–209, Feb 1983.

[12] O. Dahlberg, "Low-dispersive Leaky-wave Antennas: A Viable Approach for Fifth Generation (5G) mmWave Base Station Antennas," Master's thesis, KTH, Stockholm, Jul 2018. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-244458

[13] IEC 60153-2:2016, "Hollow metallic waveguides - Part 2: Relevant specifications for ordinary rectangular waveguides," International Electrotechnical Commission, Geneva, CH, International standard, May 2016.
[14] M. Vukomanovic, J. Vazquez-Roy, O. Quevedo-Teruel, E. Rajo-Iglesias, and Z. Sipus, "Gap waveguide leaky-wave antenna," *IEEE Transactions on Antennas and Propagation*, vol. 64, no. 5, pp. 2055–2060, Mar 2016.

Millimeter-Wave Pencil Beam Leaky-Wave Antenna

Tom Eriksson and Erik Westberg

***Abstract*—Moving into higher frequencies, in occurrence with modern applications, poses the issue of higher attenuation of electromagnetic waves, which in turn demand more directive antennas. This paper proposes a directive antenna design operating at 30 GHz based on leaky-wave technology. The model consists of two main components. Firstly, a corrugated parallel plate waveguide serves the purpose of controlling the propagation of electromagnetic waves, in particular the guided wavelength. Secondly, an array of continuous transverse stubs are implemented in the parallel plate waveguide, which allows for radiation into free space and gives a directive beam due to the array configuration. Dispersive properties of the waveguide are studied to select appropriate dimensions for the corrugations and optimization of the transverse stub dimensions is performed by a unit cell parameter analysis. The proposed design produce pencil beam radiation in the broadside direction with a gain of 24.5 dBi and a −3 dB relative bandwidth of 8.8 % and an aperture efficiency of 79 %.**

***Sammanfattning*—Att gå upp i frekvens för att möta krav satta av moderna tillämpningar för med sig problemet med högre attenuering av elektromagnetiska vågor. Detta sätter i sin tur krav på mer riktade antenner för att kompensera för förlusterna. I rapporten presenteras en riktad läckande-vågsantenn för 30 GHz. Modellen består av två huvudsakliga komponenter. Först en korrugerad parallellplåts-vågledare, vars syfte är att kontrollera hur vågen propagerar, särskilt med avseende på våglängd. Sedan en serie med transversella öppningar som tillåter utstrålning, där seriens utformning ger upphov till en direktiv stråle. Dimensioner bestäms genom dispersionsanalys av den korrugerade vågledaren och optimering av den strålande enhetscellen sker genom en parameterstudie. Den föreslagna modellen producerar en riktad stråle med antennvinsten 24.5 dBi, relativa bandbredden 8.8 % och apertureffektiviteten 79 %.**

***Index Terms*—Leaky-wave antenna (LWA), Continuous transverse stubs (CTS), Parallel plate waveguide (PPW), Dispersion.**

Supervisors: Qiao Chen and Oscar Quevedo-Teruel

TRITA number: TRITA-EECS-EX-2021:173

I. Introduction

The usage of consumer mobile data transfer is increasing rapidly, even to the point of surpassing the wired counterpart. Applications that require low latency and high broadband wireless communication are presented and developed every year. Especially advances in autonomous vehicles, industrial robots, and new generations of mobile phones are pushing the demand. The most commonly presented solution to meet the increased demand of wireless communication is to move up in frequency, into the millimeter-wave band [1]. This does pose the issue of higher attenuation which will have to be compensated for in antennas, namely more directive designs with higher efficiencies [2].

Leaky-wave antennas (LWAs) function through the feeding of a waveguide, in which a controlled leakage is implemented along the waveguide, resulting in a directive beam.

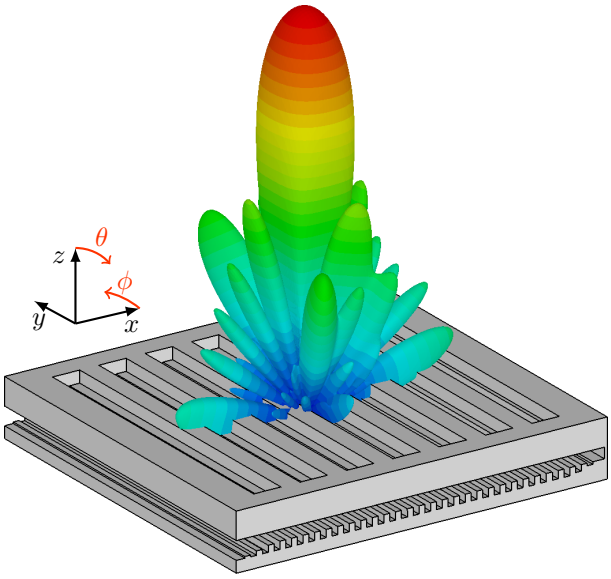


Fig. 1. 3D view of proposed design superimposed on a farfield directivity plot of the antenna at 30 GHz. Both the top plate with the CTS and the corrugated bottom plate are visible.

A fundamental attribute of leaky-wave antennas is that the direction of the beam changes as the frequency changes, called scanning [3]. This does not mean that LWAs are unsuitable for communication applications, the attributes of high gain and a low profile still prove a great potential for communication in the millimeter-band [4]. Another good candidate for high-frequency communication is phased array antennas, which are flexible but also inherently more complex requiring electronic phase shifters [5]. In comparison, a major advantage of leaky-wave antennas is their architectural simplicity.

There are generally two different types of LWAs, uniform and periodic. The uniform type typically utilize a continuous slit along the waveguide to create radiation whereas the periodic type instead implements discrete radiating elements.

One effective method to implement a periodic LWA is with the use of continuous transverse stubs (CTS), that is periodic slots in a waveguide that allows for radiation into free space. The CTS structure is readily studied. In [6], a frequency-independent radiation pattern is achieved by feeding the stubs in parallel. This is in contrast to [7] where the stubs are fed in series and the antenna is used for its frequency scanning capabilities, and in [2] where the beam direction can be altered by a planar rotation of the plate housing the slots.

This project aims to produce a periodic LWA using the CTS method in a similar fashion to the design exemplified in [2] achieving a highly directive antenna for communication purposes, but without the steering capabilities. The report is

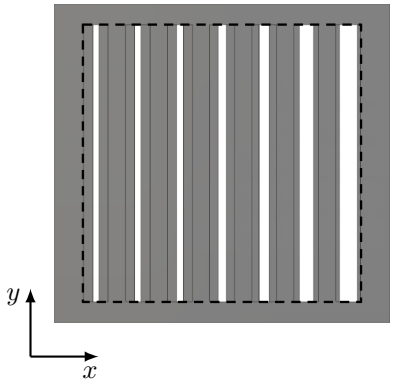


Fig. 2. Top view of antenna top plate with square aperture marked out. The CTS are visible as the vertical rectangular cutouts ranging from small to large going from left to right.

structured as follows. Section II shows some of the principles which guide the design process, followed by section III which focuses on the main building blocks of the antenna. In section IV the results are presented and discussed. Finally, a summary is provided in section V.

II. Design Principles

A. Overview

The goal of the project is to use the established building blocks from [2] and [8] to produce an antenna that generates a highly directive beam orthogonal to the antenna (broadside) in the operating band around the design frequency of 30 GHz. This article determines how to set dimensions of these building blocks to fit the design criteria without relying on excessively small details that are difficult to manufacture. For this reason, the project group imposes a limitation on the design to not use dimensions that would require a mill smaller than $1\,\mathrm{mm}$ to manufacture. An overview of the proposed design and the qualitative far-field radiation plot is presented in figure 1.

The proposed design is categorized as a 2D leaky-wave antenna, utilizing a periodic CTS structure fed in series through a parallel plate waveguide (PPW). This type of feeding poses the advantages of a high radiation efficiency and a lower profile allowing for slim designs [4], [9]. The structure consists of a bottom plate with corrugations of a specific size to control the wavelength inside the PPW, and also an upper plate that contains the radiation slots or CTS, both illustrated in figure 1. Since periodic LWAs typically struggle to produce broadside radiation, as open stopband characteristics usually appear [3], the reflection canceling technique presented in [2] is adopted, suppressing the stopband with reflection canceling notches.

The design procedure begins with studying the effects of array configurations, allowing the slot periodicity (p_1) to be set. The dimensions of the corrugated PPW are then determined to match the array configuration, followed by the determination of CTS dimensions to achieve proper illumination across the aperture. During the entire process, the concept of space harmonics provides information about the antenna's capabilities but is not used to determine any dimensions.

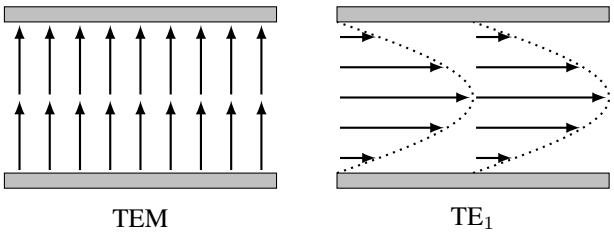


Fig. 3. Two parallel plate waveguides showing the E-field distribution of the TEM and TE_1 modes. The wave propagates into the paper.

Some other design choices include that the length of the slots is adjusted to make the aperture square which can be seen in the top-down view of the top plate in figure 2. In addition, the first three corrugation ridges in the bottom plate where the wave enters the antenna are tapered to reduce reflections of the incoming wave, and a backplate is implemented mating the top and the bottom plates of the antenna improving structural integrity. To feed the structure, the possibility of using another LWA presented in the J3a project is demonstrated.

B. Parallel plate waveguide

The main component used to guide electromagnetic waves in the antenna is a PPW, which as the name suggest consist of two metal plates in parallel to each other distanced by some dielectric material. The propagation of EM-waves inside a PPW is governed by the propagation constant γ which properties can be described by

$$\gamma = \alpha + j\beta = \sqrt{-\omega^2\mu\epsilon + \left(\frac{n\pi}{b}\right)^2} \tag{1}$$

where n is the order of the mode, ω the angular frequency and b is the height of the PPW. The constants μ and ϵ are material properties [10]. The modes describe different formats with which the EM-wave propagate and which modes that can propagate are mainly determined by the boundary conditions and dimensions of the structure. Since the sides of a PPW have open boundaries the fundamental mode will be a TEM mode (transverse electromagnetic), as in $n = 0$.

By solving (1) for $\gamma = 0$, the cutoff frequencies for the n:th modes can be calculated as

$$f_c = \frac{n}{2b\sqrt{\epsilon\mu}} \tag{2}$$

which describe where the modes can start to propagate and is mainly controlled by the height of the PPW. The first higher order modes for a PPW are the TE_1 and TM_1 (transverse electric and transverse magnetic), which both have the same cutoff frequency [10]. The electric field distributions of TEM and TE_1 is shown in figure 3.

C. Space harmonics

An important aspect when designing a periodic LWA is that only a single mode should be supported inside the structure. This mode also needs to be a slow wave, meaning that the phase constant has to be greater than the wavenumber in free space, $\beta > k_0$, and will not radiate. This is one of the main

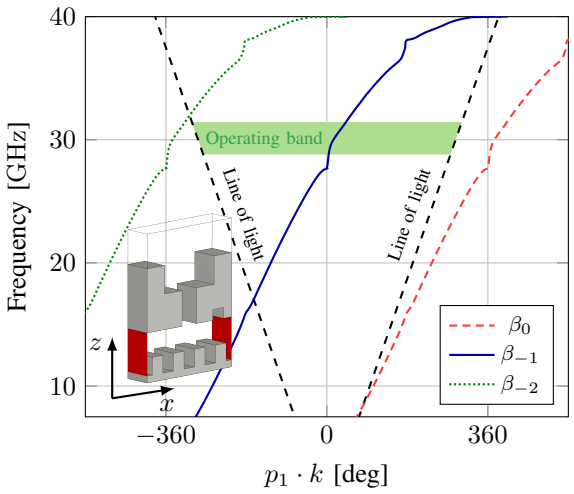


Fig. 4. Dispersion of three space harmonics corresponding to index, 0, -1 and -2. The phase, $p_1 \cdot k$, depicts the phase difference between the periodic sides of the model. Dimensions used for the model is based on the third slot of the full structure, meaning that the periodicity, $p_1 = 8\,\mathrm{mm}$.

differences from a uniform LWA for which the fundamental mode is fast and will radiate [3].

For the periodic structure, an infinite number of space harmonics are created where all harmonics together make up the fundamental mode. The phase constant, β_n of each space harmonic is separated according to

$$\beta_n = \beta_0 + \frac{2\pi}{p_1} n, \quad n = 0, \pm 1, \pm 2 \ldots \tag{3}$$

where p_1 denotes the periodicity of the elements and n the index of the harmonic [3].

For an LWA where only one beam is desired, only one of the space harmonics is allowed to be fast and radiate. In this case, the β_{-1} harmonic is chosen to radiate, which is the most common choice [3]. To show the harmonic relations, a study of the structure's dispersive properties is conducted. In this case a unit cell of the full model is used to perform the analysis, with the dispersion shown in figure 4. As it is clear that only the desired harmonic is above the line of light in the operating range, only one beam will be formed. The figure also shows that as the frequency is increased further, a second beam will appear as the β_{-2} harmonic increases above the line of light and starts to radiate.

Performing a dispersive analysis for a structure with at least one open boundary condition poses some challenges in simulation, which is why different simulation methods need to be considered. One such method uses the ABCD-matrix of a unit cell. This matrix can calculate the output of the periodic element, given the input. Using this method figure 4 is produced by applying

$$p_1 \cdot k = \arccos \frac{A(f) + D(f)}{2} \tag{4}$$

derived from [11], where A and D is calculated from the simulated S-parameters, reflections and transmissions, obtained using the *time domain solver* of CST Microwave studio. The unit-cell used matches the dimensions of the third slot of the full structure, see section III-B. Only considering the real part of k gives $p_1 \cdot \beta$ displayed as β_0 in figure 4.

D. Array

The direction of the main beam formed by radiation from the β_{-1} harmonic is given by

$$\sin(\theta) = \frac{\lambda}{\lambda_g} - \frac{\lambda}{p_1} \tag{5}$$

where λ and λ_g are the free space wavelength and the guided wavelength respectively of the fundamental space harmonic β_0 and p_1 is the periodicity of the slots [3]. Broadside radiation, corresponding to $\theta = 0$, will be achieved when the periodicity of the slots matches the guided wavelength. This requirement, $p_1 = \lambda_g$, is central for the positioning of the slots in the top plate and for the design of the corrugated bottom plate.

The radiation from an antenna array can be determined from the radiation from a unit cell and the position, phase and amplitude of the array elements. Let

$$\mathbf{k}(\theta, \phi) = \frac{2\pi}{\lambda} \begin{bmatrix} \sin(\theta)\cos(\phi) \\ \sin(\theta)\sin(\phi) \\ \cos(\theta) \end{bmatrix} \tag{6}$$

be a wave vector, $\mathbf{r}_n$ be the position of radiating element n and w_n its complex weight containing both amplitude and phase. In the case of this study, the antenna consists of an equal amplitude in-phase array which can be represented by equal real weights. The power pattern of the array $P(\theta, \phi)$ is given by

$$P(\theta, \phi) = P_0(\theta, \phi)|\Lambda(\theta, \phi)|^2 \tag{7}$$

where $P_0(\theta, \phi)$ is the power pattern of a unit cell. $\Lambda(\theta, \phi)$ is given by

$$\Lambda(\theta, \phi) = \sum_{n=1}^{N} w_n e^{-j\mathbf{k}(\theta,\phi)\cdot\mathbf{r}_n} \tag{8}$$

and is called the array factor [12], [13]. The square of the absolute value of the array factor, $|\Lambda(\theta, \phi)|^2$, for seven linearly spaced equal weight cells at 30 GHz is plotted in figure 5 for three different spacing distances.

III. Building Blocks

A. Corrugated PPW

The purpose of the parallel plate waveguide is to guide the traveling wave to each radiating slot of the top plate. As previously mentioned only one mode should be supported inside the waveguide, meaning that the height of the PPW has to be chosen to prevent the propagation of any higher-order modes within the operating frequency. Using the height, b = 3.1 mm, the cutoff for first higher-order modes, inside the air-filled PPW, are placed above the operating range with a wide margin. This means that the fundamental mode will have quasi-TEM properties.

The next design step to consider is the corrugations grooved into the lower part of the PPW which contribute to two factors. Firstly, this causes the fundamental mode to be a slow wave which is necessary as described previously. Secondly, not

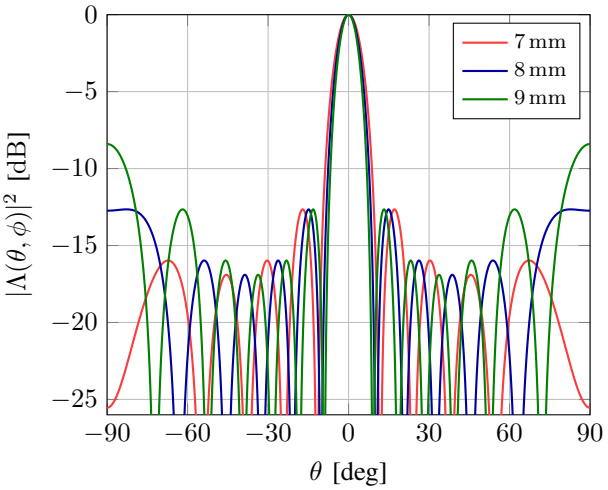


Fig. 5. Normalized array factor calculated for a wave at 30 GHz for three different slot periods, 7 mm, 8 mm and 9 mm. The angle ϕ is fixed at zero as θ is swept from $-90°$ to $90°$.

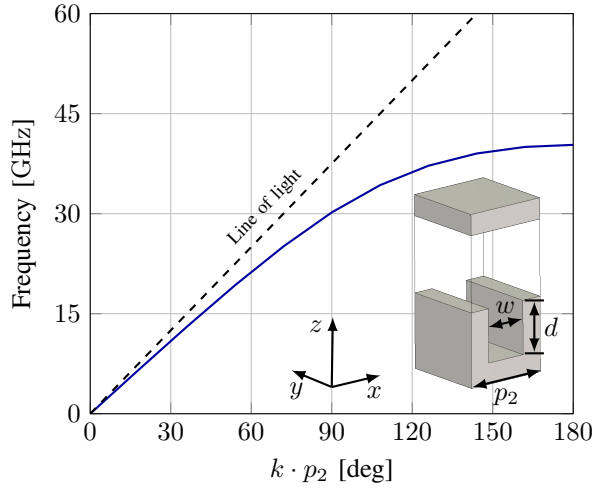


Fig. 6. Dispersive relation for a corrugated PPW, which creates a slow wave. The dimensions used are: $p_2 = 2.00\,\text{mm}$, $d = 1.28\,\text{mm}$ and $w = 1.00\,\text{mm}$.

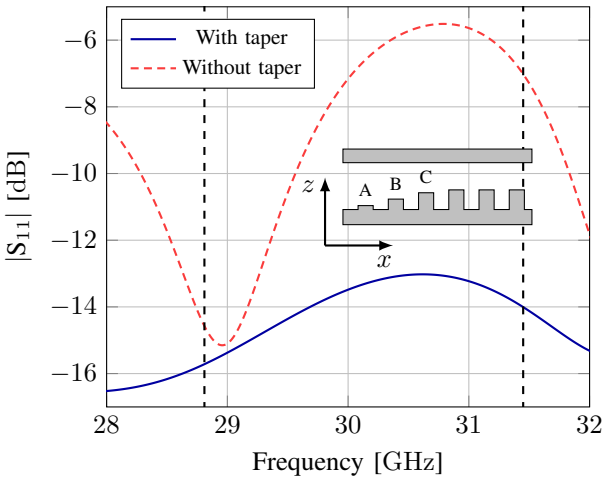


Fig. 7. Comparison between tapered and not tapered corrugations. When tapered, the height of the ridges A, B, C are 0.26 mm, 0.69 mm and 1.11 mm respectively. When not tapered they are all 1.28 mm. The operating range is marked with vertical lines.

completely unrelated, the dimensions of the corrugations will control the guided wavelength inside the PPW.

At design frequency, the guided wavelength λ_g needs to match the periodicity of the radiating slots which is chosen to 8.0 mm. A smaller value has the advantage of reducing grating lobes as demonstrated in figure 5 but has the disadvantage of imposing stricter requirements on the corrugations in the bottom plate to achieve the smaller guided wavelength. From figure 5 it is possible to see that a spacing of 8.0 mm is the largest allowed spacing without the grating lobe levels rising significantly above the first side lobe levels. According to (7) the total power pattern is a product of the unit cell power pattern and $|\Lambda(\theta,\phi)|^2$ which means that as long as either the array factor or the unit cell power pattern is small, the total radiation will be small. Combining this with the fact that the radiation from each slot $P_0(\theta,\phi)$ is somewhat directive, the grating lobes visible in the array factor will be further suppressed.

Simulations of the corrugations were done using the *eigenmode solver* in CST Microwave Studio and in figure 6 the dispersion of the PPW with corrugations is shown. Through the diagram it is shown that the corrugations form a slow wave in the structure as it is below the line of light. It also shows that the proper wavelength is created, which can be calculated from the graph using

$$k = \frac{2\pi}{\lambda_g} \implies \lambda_g = \frac{360 \cdot p_2}{phase} \tag{9}$$

where the phase, $k \cdot p_2$, at 30 GHz is very close to the desired $90°$ which give the sought wavelength of 8 mm.

Another issue of designing the corrugated PPW is the reflections induced by the wave entering the corrugations. To mitigate this the first three corrugations are tapered to specific heights, lowering the reflection significantly as displayed in figure 7. This is due to a better match of impedances resulting in a smoother transition into the corrugated PPW.

B. Radiation slots

In order to achieve uniform illumination, the dimensions of each slot have to be tapered to increasingly radiate more power relative to the ingoing power of each cell. This is simply due to that less power reaches the later slots as power leaks out at each previous slot. Each unit cell, illustrated in detail in figure 8, encompass three key features, a radiating slot, a coupling PPW (CPPW), and a reflection canceling notch.

The functionalities of the radiating slot are explained in [2] as mainly a transition from the structure to free space that better matches the impedance and cancels the complex reflection over a wide range of frequencies [14]. Also, the radiating slots allow for a reduction of the mutual coupling between elements, meaning that the contribution of each slot can be considered in isolation.

To act as a transition from the corrugated PPW to the radiation slot a CPPW is used. By controlling the height of the CPPW, h_c, a specific power reaching the radiating slot is

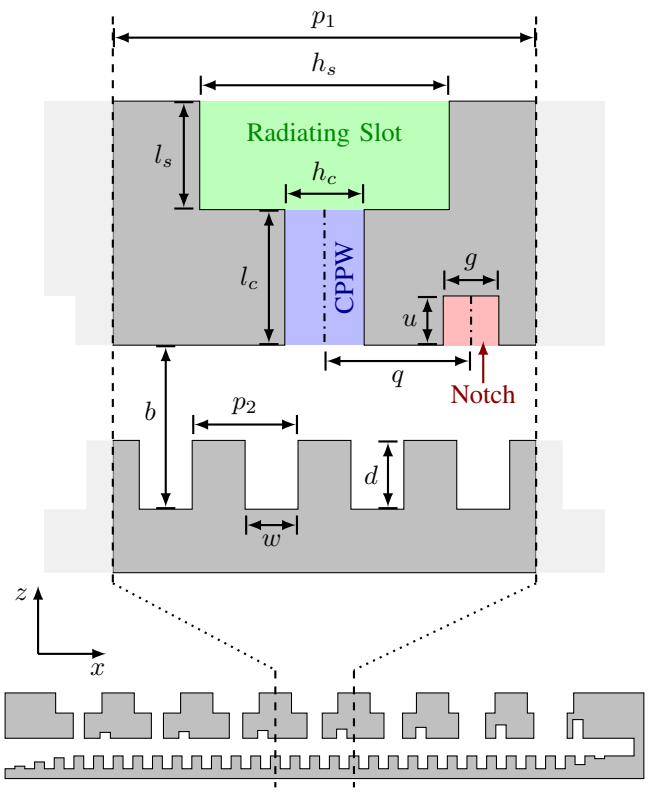


Fig. 8. Side view of the antenna with a detailed depiction of a unit cell. Dimensions used for for the unit cell are: $h_s = 4.715\,\text{mm}$, $l_s = 2.05\,\text{mm}$, $l_c = 2.56\,\text{mm}$ and $g = 1.03\,\text{mm}$. The values are acquired from [2] and rescaled for 30 GHz.

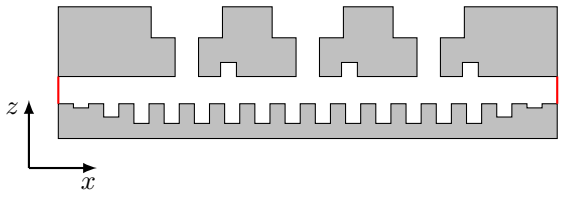


Fig. 9. Model of three identical unit cells used for calculating radiated power related to specific values of the parameters: h_c, u and q. The corrugations are tapered at the ends to reduce reflections.

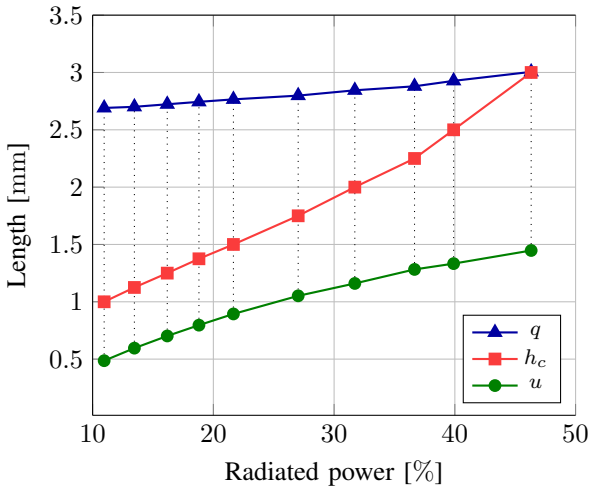


Fig. 10. Parameters of the unit cell plotted against the radiation power at design frequency. Values that cross the same vertical line are optimized relative to each other.

achieved. This is the main method used to control the radiated power of each slot to create a uniform illumination over the structure.

The purpose of the notch in the bottom plate is to reduce the reflections from the CPPW, as an implementation of the reflection canceling technique [2]. It works by creating a controlled refection from the notch which cancels out the reflection from the radiation slot. This control is achieved by firstly adjusting the height, u, of the notch which controls the power of the reflected wave. Secondly, the distance from the slot to the notch, q, controls the phase of the reflected wave which has to cancel out the reflection of the slot. Through this method, the slots can be excited in phase with each other resulting in broadside radiation.

To find the appropriate dimensions for h_c, u, and q, a smaller model consisting of three identical slots is created as shown in figure 9 and then simulated using the *time domain solver* of CST Microwave Studio. The main advantage of using three slots in the model compared to a single slot is that at least a part of the mutual coupling between the slots is included which is more representative of the function within the full structure. With a fixed dimension for h_c the dimensions for u and q are then optimized to reduce the reflection as low as possible achieving levels of S_{11} below $-25\,\text{dB}$. The relative radiation power from each slot, η, is related to the S-parameters as

$$(1-\eta)^3 = |S_{11}|^2 + |S_{21}|^2 \tag{10}$$

which when solved for η gives the radiation power as

$$\eta = 1 - \sqrt[3]{|S_{11}|^2 + |S_{21}|^2} \tag{11}$$

provided the S-parameters [9]. This results in the graph shown in figure 10 which depicts the relation between the radiation power to optimized parameters.

The final step of the slot design consists of calculating the percentage of power each slot should radiate. Aiming for $95\,\%$ total radiation results in a per slot radiation, P_{rad}, of $13.57\,\%$ of the total power. As previously mentioned this requires the relative power radiation of each specific slot, η_n, to be increased as the remaining power decreases at each slot. This can be calculated by solving

$$P_{rad} = \eta_n \prod_{1 \le i < n} (1 - \eta_i) \tag{12}$$

as exemplified in [9].

Interpolating the values in figure 10 for each η_n give the dimensions used for the unit cells, as is shown in table I. The only exception to the method is the design of the last slot, index seven, which has too extreme dimensions and therefore is optimized separately to give the proper radiated power.

C. Feeding structure

To feed the structure another LWA presented in project J3a can be used. Their design consist of a more traditional LWA

TABLE I
SELECTED VALUES OF RELATIVE RADIATED POWER, η, AND GEOMETRICAL DIMENSIONS FOR UNIFORM ILLUMINANCE.

index	1	2	3	4	5	6	7
η [%]	13.6	15.7	18.6	22.9	29.7	42.2	73.1
h_c [mm]	1.13	1.23	1.37	1.55	1.89	2.70	3.40
q [mm]	2.70	2.72	2.74	2.77	2.83	2.97	2.80
u [mm]	0.60	0.68	0.79	0.93	1.11	1.36	1.90

using a WR34 rectangular waveguide, with one side opened up replaced by periodic pins which are tapered to provide sinusoidal illumination. At 30 GHz their structure has highly directive radiation which can be fed into the CTS structure by guiding the radiation with a PPW. An illustration of the combination between the models is shown in figure 11.

IV. RESULTS AND DISCUSSION

As the structure proposed in this report does not have a standard feeding input without the use of the design proposed in project J3a, the independent performance of the structure is difficult to measure physically. The results used to evaluate the performance are taken from simulations made with CST Microwave Studio (release version 2020.00 - Sep 25, 2019).

A. Radiation properties

The radiation pattern for the independent structure is shown in figure 12, also depicting a comparison to the optimal calculated pattern. Directivity represents how directive the antenna is, in comparison to the directivity of a fully isotropic antenna. The calculations are made by applying the array factor discussed in section II-D to the simulated results of a unit cell. The unit cell power pattern's dependence on the height of the CPPW and the notch dimensions are low and not taken into explicit account. The results show that at design frequency the structure achieves the main lobe directivity of 24.6 dBi with side lobe levels (SLL) of -14.79 dB.

The gain of the antenna, which is similar to directivity but also accounts for losses, is simulated to 24.5 dBi at 30 GHz in the broadside direction. In figure 13 the gain in the broadside direction is shown over a range of frequencies, resulting in a -3 dB bandwidth of 2.64 GHz which equates to a relative bandwidth of 8.8 %.

Aperture efficiency describes how efficiently the antenna's radiating area is used. This results in an efficiency of 78.8 % by comparing the physical aperture seen in figure 2 to the effective aperture A_e derived from the simulations with

$$A_e = \frac{\lambda^2}{4\pi} G \tag{13}$$

where G is the gain [10].

One of the drawbacks of using an LWA for fixed direction communication purposes becomes apparent with the bandwidth, as the main limitation comes from the beam moving away from the broadside due to the scanning attribute. This is a well-known drawback and reducing the scanning is one of

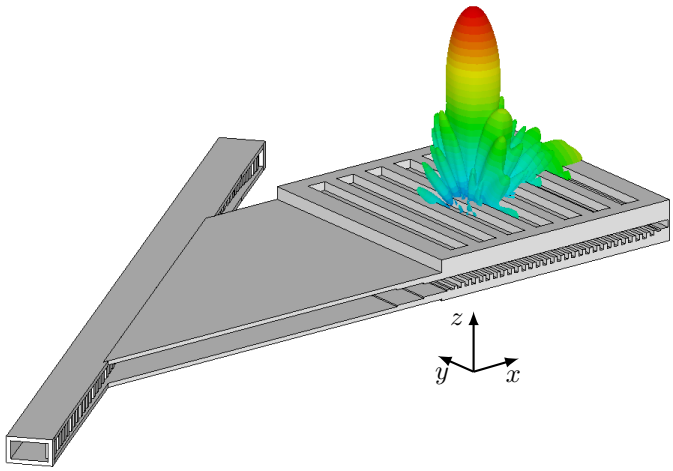


Fig. 11. A combination of the two models represented in projects J3a and J3b. A PPW matches the heights of the two components with two small steps. The farfield directivity which they form together is displayed.

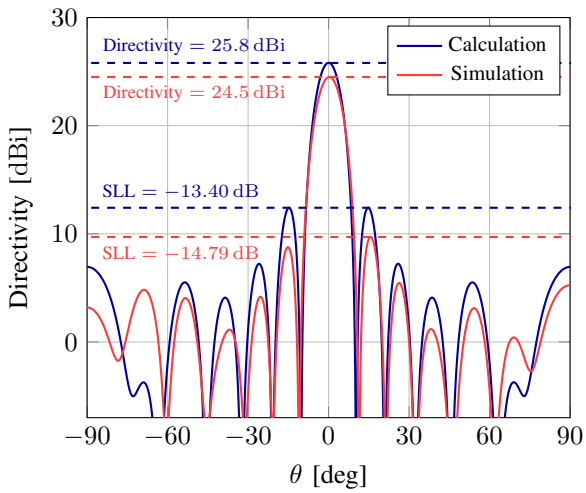


Fig. 12. Directivity of the antenna compared to calculated pattern. The angle ϕ is fixed at zero as θ is swept from $-90°$ to $90°$.

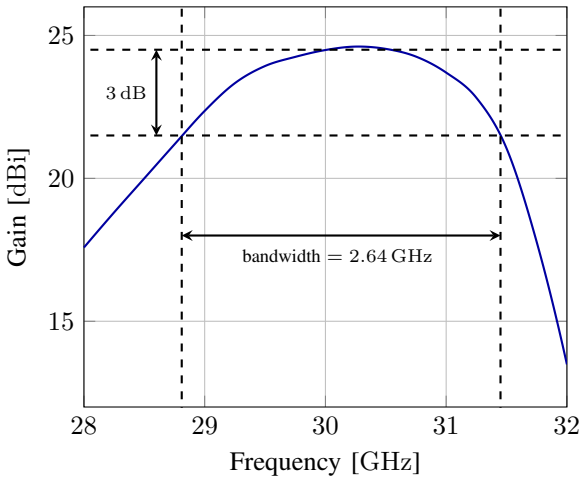


Fig. 13. Simulated gain for a range of frequencies. The gain is measured in the broadside direction. The -3 dB bandwidth has its reference at 30 GHz.

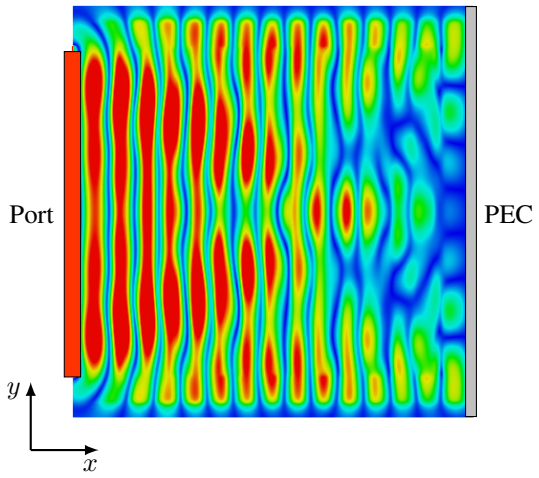


Fig. 14. Cutting plane of the structure, showing the absolute value of the E-field inside the PPW at 30 GHz.

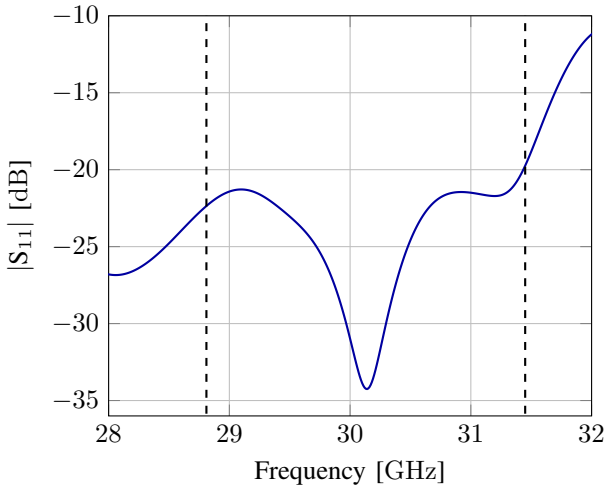


Fig. 15. The reflections in terms of the S_{11} parameter plotted against frequency, where the bandwidth is marked with vertical lines.

the main areas in which LWAs should be further researched for communication applications.

The discrepancies between results and calculations can be explained mainly due to two different factors. Firstly, by qualitatively studying the E-field inside the structure, displayed in figure 14, it is clearly shown that the illumination along the y-axis is not completely uniform inside the structure. This is due to leakage and reflections induced on the sides of the structure, resulting in interference. Since the calculations are based on a model which does have a uniform illumination, it is reasonable that the results show a slightly lower directivity. To improve the structure in this regard a stopband structure could be explored and applied around the structure as it is done in [2].

Secondly, the reason behind the lower SLL can be explained by the reflections induced by the final radiating slot which was primarily optimized for high radiation, which is difficult to combine with low reflections. This causes the earlier slots to radiate slightly more than calculated, which causes the SLL to diverge from expectations. A very small amount of these reflections are also induced by ending the structure with a conducting wall. A small improvement should therefore be feasible by ending the structure with a matched load [3] reducing said reflection. However, since the SLL is lower than calculated, it is not a problem but rather a discrepancy from the expected results.

B. Reflections

The reflections of the structure are given by the S_{11} parameter, displayed in figure 15. The reflections within the operating band remain under $-20\,\mathrm{dB}$. This result could also be improved by ending the structure with a matched load, but as the reflections already are low the difference would be minimal.

C. Consolidation with feeding structure

To verify that the combination of models, as displayed in figure 11, actually works a simulation has been performed. This results in reflections lower than $-15\,\mathrm{dB}$ over the operating band, as well as the directivity of $22.9\,\mathrm{dBi}$ at broadside of the CTS antenna. Together these results confirm that combining the models this way is possible. To further optimize this combination, the sizes of the models can be altered to better fit each other increasing the power transmitted into the CTS structure.

As the combination is currently executed it proves no real purpose compared to feeding more conventionally. However, as future work, it could be studied how the CTS structure can be adjusted to accept the radiation from the 1D LWA at a multitude of angles. This has the potential to produce a pencil-beam antenna that can scan in multiple directions. A way to achieve this may be to change corrugations to a structure of similar properties, e.g a bed of nails. To continue work on the project, both the designs presented in this paper and in J3a should be constructed through computer-controlled milling and tested together to further prove the functionality.

V. Conclusions

A 2D leaky-wave antenna designed for $30\,\mathrm{GHz}$ has been presented. The design utilizes CTS functionality to control the radiation patterns and a corrugated PPW to feed the structure with a controlled wavelength. The results show a highly directive beam in the broadside direction with a peak directivity of $24.6\,\mathrm{dBi}$. The design is shown to be effective over the relative bandwidth of $8.8\,\%$ with high efficiencies in both aperture and reflection at the design frequency. Combining the structure with another structure, presented in project J3a, is proven possible and shows potential for more functionalities.

Acknowledgment

The project group expresses special thanks to Qiao Chen for his supervision and guidance of the project. The group also gives their gratitude to Oscar Quevedo-Teruel for providing relevant lectures and structuring the projects of the J context.

REFERENCES

[1] W. Hong, Z. H. Jiang, C. Yu, D. Hou, H. Wang, C. Guo, Y. Hu, L. Kuai, Y. Yu, Z. Jiang, Z. Chen, J. Chen, Z. Yu, J. Zhai, N. Zhang, L. Tian, F. Wu, G. Yang, Z.-C. Hao, and J. Y. Zhou, "The role of millimeter-wave technologies in 5g/6g wireless communications," *IEEE Journal of Microwaves*, vol. 1, no. 1, pp. 101–122, 2021.

[2] K. Tekkouk, J. Hirokawa, R. Sauleau, and M. Ando, "Wideband and large coverage continuous beam steering antenna in the 60-ghz band," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 9, pp. 4418–4426, Sep. 2017.

[3] J. L. Colakis, "Leaky-wave antennas," in *Antenna engineering handbook*, forth edition ed. New York: McGraw-Hill, 2007.

[4] Q. Chen, O. Zetterstrom, E. Pucci, A. Palomares-Caballero, P. Padilla, and O. Quevedo-Teruel, "Glide-symmetric holey leaky-wave antenna with low dispersion for 60 ghz point-to-point communications," *IEEE Transactions on Antennas and Propagation*, vol. 68, no. 3, pp. 1925–1936, Mar. 2020.

[5] J. L. Colakis, "Phased arrays," in *Antenna engineering handbook*, forth edition ed. New York: McGraw-Hill, 2007.

[6] M. Ettorre, F. F. Manzillo, M. Casaletti, R. Sauleau, L. Le Coq, and N. Capet, "Continuous transverse stub array for ka-band applications," *IEEE Transactions on Antennas and Propagation*, vol. 63, no. 11, pp. 4792–4800, 2015.

[7] Y. You, Y. Lu, Q. You, Y. Wang, J. Huang, and M. J. Lancaster, "Millimeter-wave high-gain frequency-scanned antenna based on waveguide continuous transverse stubs," *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 11, pp. 6370–6375, 2018.

[8] W. W. Milroy, "Continuoustransverse stub element devices for flat plate antenna arrays," U.S. Patent 5 483 248, Jan. 1996.

[9] R. S. Hao, Y. J. Cheng, and Y. F. Wu, "Shared-aperture variable inclination continuous transverse stub antenna working at k- and ka-bands for mobile satellite communication," *IEEE Transactions on Antennas and Propagation*, vol. 68, no. 9, pp. 6656–6666, 2020.

[10] D. K. Cheng, *Field and wave electromagnetics*, second edition ed. Harlow, Essex, GB: Pearson, 2014.

[11] G. Valerio, Z. Sipus, A. Grbic, and O. Quevedo-Teruel, "Accurate equivalent-circuit descriptions of thin glide-symmetric corrugated metasurfaces," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 5, pp. 2695–2700, 2017.

[12] S. Silver, H. M. James, J. E. Eaton, L. J. Eyges, T. Keary, H. Krutter, G. G. Macfarlane, R. Redheffer, J. Risser, S. Silver, O. A. Tyson, and L. Van Atta, "Pattern theory," in *Microwave Antenna Theory and Design*, first edition ed. New York: McGraw-Hill, 1949, pp. 258–261.

[13] J. B. Peter. (2021, April) The array factor. antenna-theory.com. [Online]. Available: https://antenna-theory.com/arrays/arrayfactor.php

[14] W. W. Milroy, "Planar antenna radiating structure having quasi-scan, frequency-independent driving-point impedance," U.S. Patent 5 995 055, Nov. 1999.

Measurement of the Complex Permittivity of Phantoms for 5G/6G Compliance Tests

William Utriainen and Adel Alizadeh

***Abstract*—In late 2019, 5G, the next generation of network technology was first deployed. The first deployment covers only a small part of the total data transfer needs. In order to increase data transfer, the frequency band used, must be increased towards the millimeter wave.**

As the frequency band is increased, safety studies must be done. The amount of radiation put out by the new 5G technology must adhere to certain standards.

This paper was written in conjunction with a study made by a group of students in KTH. The aim of the project, was to determine the amount of radiation absorbed from the new 5G technologies. Thereby also determining if the radiation is permissible according to safety standards.

The method by which the study achieves its aim is, by an approach called 'heating kinetics'. Heating kinetics means that, the absorbed radiation in a sample (phantom), is determined from its temperature increase.

The group of students was divided into two sub-groups. Each sub-group being assigned different parts of the project. This paper shows the work of one of the sub-groups.

The work involves determining material properties of the radiated sample (phantom) through simulation and experimental measurements. The result is a complex permittivity. The conclusion of the study is that the method is a valid way of obtaining material properties.

***Sammanfattning*—I slutet av 2019 implementerades nästa generation av nätverks-teknologi, 5G. Den första implementationen täcker endast en liten del av det totala behovet av dataöverföring. För att öka dataöverföringen måste frekvensbandet utökas för att inkludera millimetervågor.**

När frekvensbandet ökas måste säkerhetsstudier göras. Människokroppen absorberar strålning i form av värme. Mängden absorberad värme måste följa vissa riktlinjer.

Denna uppsats skrevs i samband med en studie gjord av en grupp studenter på KTH. Syftet med studien var att bestämma mängden strålning från den nya 5G-teknologin. Därmed avgörs även om strålningen faller inom ramen av säkerhets-standarder.

Metoden med vilken studien uppnår sitt syfte kallas 'uppvärmnings-kinetik'. Uppvärmnings-kinetik innebär att den absorberade strålningen i ett prov bestäms från dess temperaturökning.

Gruppen av studenter delades in i två undergrupper. Varje undergrupp hade varsin ansvars-tilldelning. Denna rapport redovisar arbetet från en av undergrupperna.

Arbetet innefattar bestämingen av material-egenskaper hos ett alstrat sampel i en simuleringsmiljö samt via experimentella mätningar. Resultaten av studien är en komplex permittivitet. Slutsatsen av studien är att metoden fungerar för att bestämma material-egenskaper.

***Index Terms*—5G, radiation, absorption, safety.**

Supervisors: Oscar Quevedo-Teruel, Oskar Zetterström, Freysteinn Vidar Vidarsson

TRITA number: TRITA-EECS-EX-2021:174

I. Introduction

The next generation of network technology (5G) has started to be deployed. The technology promises greater communication capabilities, which are needed and are described in [1]. Safety testing must be done, to ensure low levels of radiation exposure, according to safety regulations in [2]. One part of safety testing, is to determine material properties of samples, hereafter referred to as phantoms. The purpose of this paper is to verify a method of determining material properties of phantoms. Previous work on the problem has been done. The reader may wish to explore the following references. Some different methods of safety testing are: use of a coaxial cable described in [3] and [4], use of a Debye-model described in [5], use of freespace technique on human skin samples detailed in [6] and use of reflection measurements detailed in [7]. One way of safety testing, which we will focus on here, is by heating kinetics as has been done by [4]. In the process described, there is one particular step, where a parameter (complex permittivity) has to be determined. This paper will show both how the value was determined and what value was obtained. As this paper was written in light of replication, it has a high degree of similarity in terms of methodology. This paper is about the results of one of the sub-groups of the study. Benefits of the described method include the following: lowered cost, as expensive millimeter wave network analyzers are not needed. Thus, it gives another way of estimating material properties. The paper is structured in the following manner: Section II goes through the theory. Section III goes through the method. Section IV goes through the results. Section V gives a discussion. Section VI goes through the conclusion.

II. Theory

The overarching steps are as follows:

1. Manufacture a phantom.
2. Radiate the phantom, while recording the temperature.
3. From the temperature increase, obtain penetration depth.
4. From the penetration depth, calculate the complex permittivity.

1) Manufacture a test phantom

A phantom was made. The phantom was made from gelatin and water.

2) Radiate the phantom, while recording the temperature

The phantom was radiated in an anechoic chamber to mimic more directive conditions, while the temperature was recorded. The recording was done with a thermal imaging camera, FLIR A65SC. See Fig. 1.

3) From the temperature increase, obtain penetration depth

From [4] we have the following partial differential equation, so called bio-heat equation:

$$\rho C \cdot \frac{\partial T_t}{\partial t} = k \cdot \frac{\partial^2 T_t}{\partial z^2} - V_s \cdot (T_t - T_b) + Q(z,t) \qquad (1)$$

Where ρ is the mass density, C is the specific heat of the exposed material, T_t is the temperature of the phantom, t is time, k is the heat conduction coefficient, z is the one dimensional position vector in the phantom, $V_s = f_b \cdot \rho_b \cdot C_b$ (which is zero because the blood flow rate $f_b = 0$ in a phantom.), T_b is the arterial blood temperature and $Q(z,t)$ is the heat deposition, which can be defined as:

$$Q(z,t) = \frac{2 \cdot (1-R)}{\delta} \cdot I_o \cdot e^{-2z/\delta} \cdot u(t) \qquad (2)$$

Where R is the power reflection coefficient, I_o is the incident power density, δ is the skin depth, z is the one dimensional position vector in the phantom, and $u(t)$ is the unit step function.

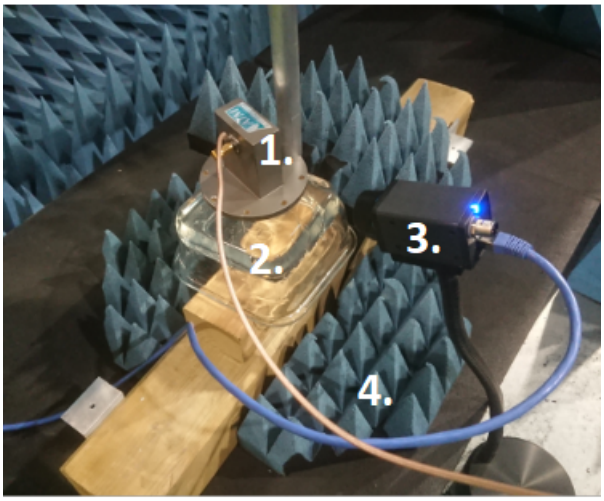


Fig. 1. The experimental setup. The antenna is marked in the figure as "1.". The phantom is marked in the figure as "2.". The camera is marked in the figure as "3.". In the background is seen blue cones of the anechoic chamber, marked "4.".

Equation (1), has the following solution, seen in [4]:

$$\begin{aligned} T(t) = \frac{q}{\lambda - 1/L^2}\{1 &+ \frac{1/L+\alpha}{2(\sqrt{\lambda}-\alpha)} \cdot \operatorname{erfc}\left(\sqrt{\frac{\lambda t}{\mu}}\right) \\ &- \frac{(1/L+\alpha)}{2(\sqrt{\lambda}+\alpha)} \cdot \operatorname{erfc}\left(-\sqrt{\frac{\lambda t}{\mu}}\right) \\ &- \frac{1}{2}\exp\left(\frac{t}{\mu}\left(\frac{1}{L^2}-\lambda\right)\right) \cdot \operatorname{erfc}\left(\frac{1}{L}\sqrt{t/\mu}\right) \\ &+ \frac{\alpha\left(\lambda - 1/L^2\right)}{(\lambda-\alpha^2)\left(\frac{1}{L}-\alpha\right)} \cdot \exp\left(\left(\alpha^2-\lambda\right)\frac{t}{\mu}\right) \cdot \operatorname{erfc}(\alpha\sqrt{t/\mu}) \\ &- \frac{(1/L+\alpha)}{2(1/L-\alpha)} \cdot \exp\left(\frac{t}{\mu}\left(1/L^2-\lambda\right)\right) \cdot \operatorname{erfc}\left(\frac{1}{L}\sqrt{t/\mu}\right)\} \\ &+ \frac{\alpha\left(T_e - T_b\right)}{\alpha + \sqrt{\lambda}} \end{aligned} \qquad (3)$$

Where $q = 2(1-R)I_o/(\delta \cdot k)$, $I = (1-R)I_o$ (referred to as the absorbed power density) and $L = \delta/2$. Equation (3) can be solved analytically. Values are available for all parameters except q and L. If values are provided for q and L, then equation (3) reduces to a single variable curve. T as a function of t. Given a temperature curve from for example measurements (T as a function of t), an optimization can be performed to find values for q and L, that match the curve to the measurements. The method by which values could be achieved from given data, which is done here, is by use of a non-linear least squares fit. (Which is implemented MATLAB.) The values for δ and I were calculated from the estimated values for q and L, giving an estimation of δ and I. Values are presented in the results section.

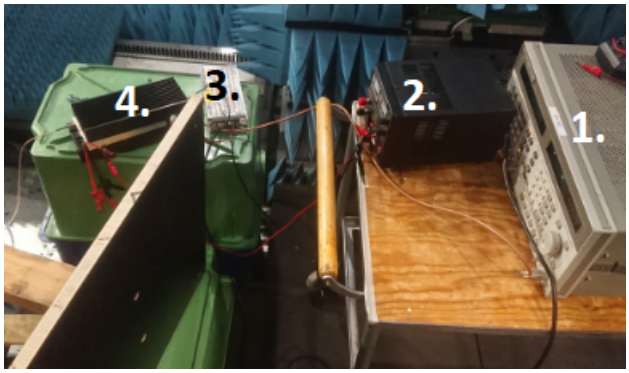


Fig. 2. Further experimental setup. The signal generator is marked in the figure as "1.". The power-supply is marked in the figure as "2.". The AC/DC-converter is marked in the figure as "3.". The radio frequency power amplifier is marked in the figure as "4.".

4) From the penetration depth, calculate the permittivity

We have the following equations from [4], known as the Debye-model:

$$\delta = \frac{c}{2\pi \cdot f \cdot \operatorname{Im}(\sqrt{\varepsilon *})} \qquad (4)$$

$$\varepsilon(\omega)^* = \varepsilon' - j\varepsilon'' = \varepsilon_\infty + \frac{\Delta\varepsilon}{1+j\omega\tau} + \frac{\sigma}{j\omega\varepsilon_0} \qquad (5)$$

Where c is the speed of light, f is the frequency of radiation, ε^* is the complex permittivity, ε_∞ is the optical permittivity, ε_s is the static (low frequency) permittivity, $\Delta\varepsilon = \varepsilon_s - \varepsilon_\infty$ is the magnitude of the dispersion of the free water fraction of the sample, j is the imaginary unit, $\omega = 2\pi f$ is angular frequency, τ is the relaxation time of free water, σ is the ionic conductivity and $\varepsilon_0 = 8.85 \times 10^{-12}$ F/m. The adjustment to the polarization due to an applied external electric field is referred to as dielectric relaxation and the Debye-model in equation (5) is a response to the dielectric relaxation in the frequency domain. The dielectric dispersion can be seen in the dependency of the complex permittivity on the frequency of the applied field. A more detailed description of the Debye-model and its constituent parameters can be seen in [8].

Note that in equation (4) the only unknown is the sought complex permittivity ε^*. The value for ε^* is found in the following manner. The skin depth was acquired from the fitting procedure, this corresponds to δ in equation (4). The values for

τ and $\Delta\varepsilon$ in equation (5) are varied and placed in equation (4) until equation (4) holds. That gives the value for the complex permittivity ε^*.

5) heat transfer coefficient

The heat transfer coefficient is included in equation (3) and gives an indication of the total heat loss to the surrounding environment due to convection, radiation and evaporation which is described in [4]. The following three equations from [9] describe the calculation of the heat transfer coefficient. The difference between h_r and h_c, is that h_r describes heat transfer by radiation, whereas h_c accounts for natural convection of air in the surrounding environment. See reference [9] for further parameter explanation.

$$h = h_r + h_c \tag{6}$$

$$h_r = \sigma_{\mathrm{SB}} \left(T + T_{\mathrm{ext}}\right) \left(T^2 + T_{\mathrm{ext}}^2\right) \tag{7}$$

$$\frac{h_c L}{k_f} = 0.59 \left[\left(\frac{L^3 \rho_f^2 \beta_f g \Delta T}{\mu_f^2} \right) \times \left(\frac{C_f \mu_f}{k_f} \right) \right]^{0.25} \tag{8}$$

III. Method

The method is divided into two sections, simulation and experimental measurements. The mathematical procedure involving the curve fitting is the same for both parts. The description involving simulation is more detailed because it highlights factors that have to be defined, such as boundary conditions.

A. Simulation

Simulation were carried out in CST microwave solver by coupling electromagnetic simulation to a thermal solver. The simulation requires a material to be constructed with defined thermal properties. Here we define a slab of material with an electric loss tangent $\tan(\delta_e) = \epsilon'/\epsilon'' = 20/74$. This corresponds to the complex permittivity of water at 5 GHz found in [4]. The thermal properties can be found in Table I. The surrounding material was set to air at an ambient temperature of 20°C. The tangential electric field along the x-direction was set to zero and the tangential magnetic field along the y-direction was also set to zero whilst the z-direction was set to open, this defines the boundary conditions. The slab of material can be seen in Fig. 3.

A right hand coordinate system is used. Fig. 4 shows a red arrow pointing to the negative z-direction, this corresponds to the direction of the propagating plane wave at a semi-infinite distance from the phantom. Note that the three arrows in Fig. 4 are not the Cartesian axes. The aforementioned boundary conditions ensure a propagating plane wave inside the phantom, a more detailed description of plane waves can be found in [10]. The electric field amplitude was set to 3000 V/m. The dimensions of the phantom were set, partly due to reducing the duration of simulation, to 20 mm in the x-direction, 20 mm in the y-direction and 40 mm in the z-direction.

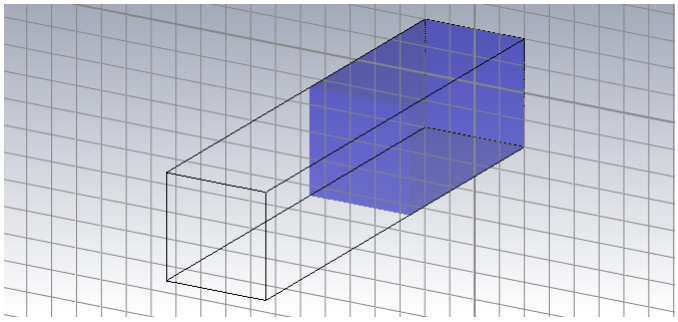


Fig. 3. Perspective view of the phantom in CST.

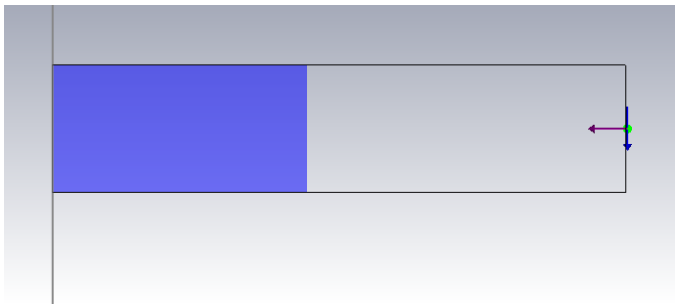


Fig. 4. Phantom in CST with the defined direction of the incident plane wave.

Fig. 5 shows the propagating plane wave in air and inside the phantom. It can clearly be seen how the wave gets attenuated while propagating inside the phantom. Moreover, this relates to the skin depth. The reference value for the skin depth was established from Fig. 6.

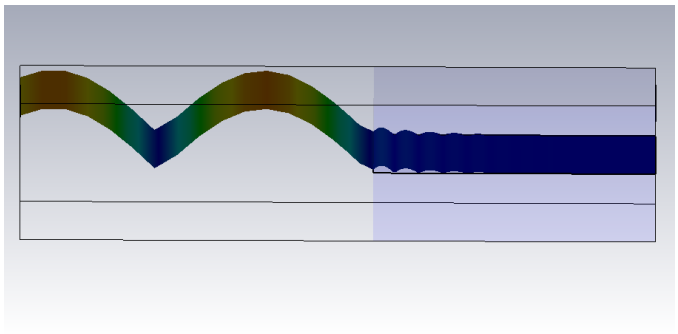


Fig. 5. Carpet view of the attenuated electric field inside the phantom.

This was done by determining the distance at which the electric field amplitude decreased by a factor $1/e$ [11]. The black line in Fig. 6 corresponds to the skin depth, $\delta_{ref} \approx 8.17$ mm. The specified thermal simulation is a transient solver and it calculates the heat distribution in the phantom by importing losses from the electromagnetic simulation, this corresponds to an electromagnetic-thermal coupled simulation as is stated in [12].

Furthermore, the boundary conditions along the x -and y-direction in the thermal simulation were set to adiabatic which corresponds to zero heat flow through a boundary, this is illustrated in [12]. The boundary condition along the z-direction was set to open. Thermal losses due to radiation (from a surface) was set by defining an emissivity, illustrated in [12]. Thermal

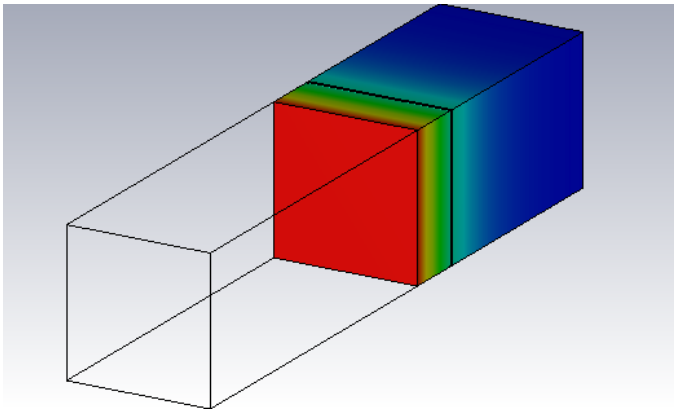


Fig. 6. Maximal electric field amplitude for determining a reference value for the skin depth.

TABLE I
THERMAL -AND ELECTROMAGNETIC PROPERTIES OF SIMULATED PHANTOM IN CST

ρ, kg/m^3	C, J/(kg $\cdot K$)	k, W/(m $\cdot K$)	h, W/(m^2 $\cdot K$)	ϵ^*
1000	3700	0.66	17.75	$74 - j20$

losses due to convection (from a surface) was set by defining a convective heat transfer coefficient h_c, as is described in [12]. The value for h given in Table I was used in CST. That value corresponds to a convective heat transfer coefficient $h_c = 12$ and a radiative heat transfer coefficient $h_r \approx 5.75$. The radiative heat transfer coefficient was calculated from equation (7). The results of the thermal simulation is presented in section IV.

B. Experimental measurements

The experimental setup can be seen in Fig. 1 and Fig. 2. The signal generator (HP8665B) was set to 4.705 GHz and these signals were further amplified by a radio frequency power amplifier. Fig 2. shows the corresponding power supplies. The phantom seen in Fig 1. consists of a mixture of approximately 15 g of gelatin and 500 g of pure water with an approximate height of 3 cm, this corresponds to a phantom consisting of 97% water. The emissivity of the IR-camera was set to 0.96. The thermal properties used in the fitting procedure can be seen in Table II.

TABLE II
ASSUMED THERMAL PROPERTIES OF MANUFACTURED PHANTOM

ρ, kg/m^3	C, J/(kg $\cdot K$)	k, W/(m $\cdot K$)	h, W/(m^2 $\cdot K$)
1000	3700	0.66	8.72

The thermal properties in Table II were taken from [9] with basis on the high water content of the phantom. As these are assumed thermal properties, the implications of these values are further discussed in section V. The results of the experimental measurements can be seen in section IV.

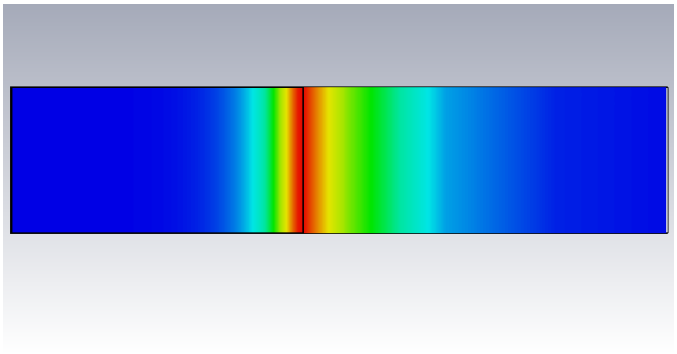


Fig. 7. Heat distribution in phantom after a duration of 10 seconds.

IV. RESULTS

The results are divided into two parts, simulation and measurements. The procedure in which the results are obtained are the same with differences in the respective analysis and discussion.

A. Simulation

The electromagnetic-thermal coupled simulations were done in CST Studio Suite. A prerequisite for estimating the skin depth of a constructed material with defined thermal properties is to first acquire the heat distribution. Fig. 7 shows the temperature distribution after 10 seconds in CST. The plane wave is incident to the surface of the material in the negative z-direction, thus the left part of the black line represents the temperature distribution in the material whilst the right part represents the surrounding material which is air. Fig. 7 clearly shows how the heat transfers in the material, thus increasing the temperature over time. Furthermore, it can be seen that the temperature of the surrounding material increases. This corresponds to air heating up close to the surface of the material.

Fig. 8 shows the surface temperature increase of the material over 10 seconds. By examining two points between 0 and 2 seconds and comparing the temperature rise to the respective increase in temperature between 8 and 10 seconds, it can be noted that the rate of temperature increase is slightly higher in the beginning. This is indicative of the expected characteristic of the curve, a more definitive example can be seen in [4].

Fig. 9 shows the result of the fitting procedure for equation (3). That is, fitting the analytical solution to the bio-heat equation is fitted to the data from CST by using a non-linear least squares method. The result from the fitting procedure is the estimated skin depth δ and absorbed power density I. By observing Fig. 9 it can be noted that the fitted curve follows the data from CST quite well, indicating the validity in the estimated parameters. As was mentioned in section III, the estimated skin depth is used alongside the Debye-model to estimate the complex permittivity ϵ^*. The estimated parameters are given in Table III. The expected value for the complex permittivity of the constructed material in CST follows from defining the electric loss tangent $\tan(\delta_e)$. This is with reference to the complex permittivity of pure water at 5 GHz, which can be found in [4]. Moreover, reference values for $\Delta\epsilon$ and τ for pure water can be found in [8], this is with respect to a wider

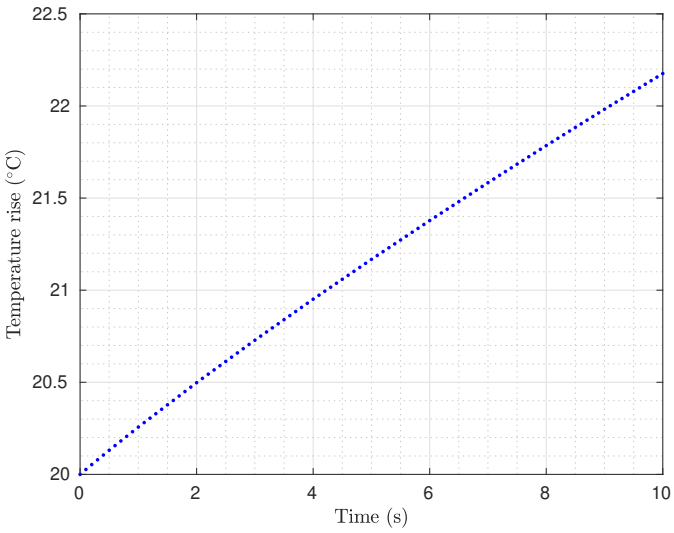


Fig. 8. Curve showing temperature rise on surface of the phantom within 10 seconds in CST.

frequency range and discrete temperature values. The values for $\Delta\epsilon$ and τ in Table III resemble those in [8].

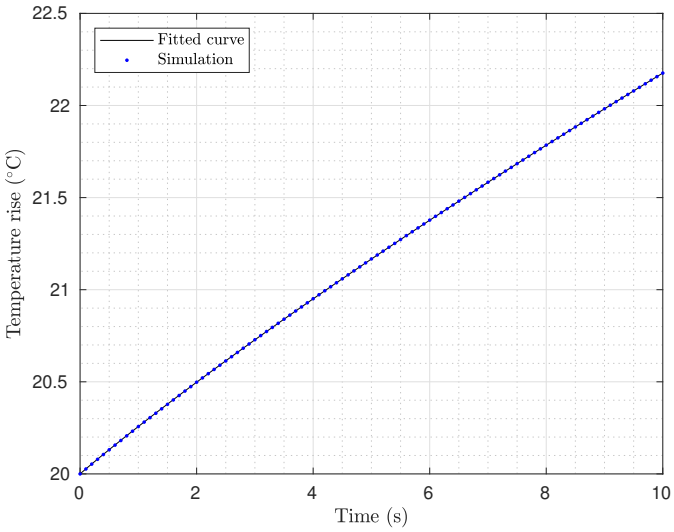


Fig. 9. The solid line shows the analytical solution to the bio-heat equation fitted to the data from the simulation.

TABLE III
ESTIMATED ELECTROMAGNETIC PROPERTIES OF SIMULATED PHANTOM IN CST

δ, mm	I, W/m^2	$\Delta\epsilon$	τ, ps	ϵ^*
8.2	4254	74.8	9.3	$74.12 - j20.14$

The reference value for the skin depth $\delta_{ref} = 8.17mm$ was given in section III, this approximately corresponds to a 0.37% difference compared to the estimated skin depth acquired from the fitting procedure. The values for $\Delta\epsilon$ and τ found in [8] are given as $\Delta\epsilon = 73.97$ and $\tau = 9.40$ ps for 20° C in the frequency range 0 - 30 GHz. Thus, the complex permittivity obtained from the Debye-model is very close to the expected complex permittivity of $\epsilon^* = 74 - j20$ of the constructed material in CST. As the fitting procedure resulted in estimations for q and L from which δ and I follow, a 95% confidence interval for q is given by $(1.5678 \times 10^6, 1.5696 \times 10^6)$ whilst a 95% confidence interval for L is given by $(0.004087, 0.004114)$. The estimated values from the fitting procedure is given by $q = 1.5686 \times 10^6$ W/m^2 and $L = 0.004109$ mm. Moreover, the mean squared error is $1.6109{\times}10^{-7}$ and the average value for the residuals is given by -$1.7624{\times}10^{-4}$. The negative value indicates a slight undershoot, nevertheless because the value is close to zero it indicates a good fit.

B. Experimental measurements

Just as was done with simulation, where a curve fitting was done for equation (3), so too, the same thing was done for an experimentally measured temperature curve. A phantom was heated up a with a wave-guide similar to the simulation setup. See Fig. 1. for an image of the setup. Fig. 10 shows a heat map of the temperature distribution of the manufactured phantom. It shows the heated spot under the waveguide captured by the IR-camera.

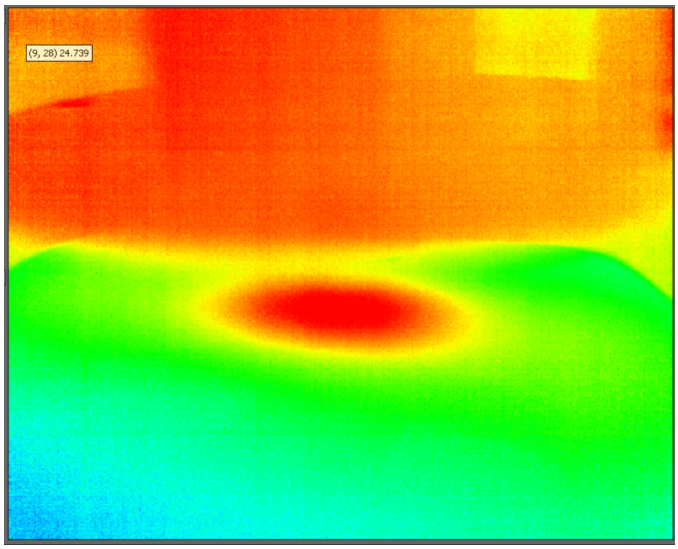


Fig. 10. Heat distribution captured by the IR-camera which clearly shows the heated spot on the phantom under the waveguide.

The IR-camera captured the surface temperature increase over approximately two minutes. The temperature rise can be seen in Fig. 11. The measurement data fluctuates and does not have the consistent rate of change of temperature as can be seen in Fig. 8. Furthermore, the curve does not have quite the same characteristics of the measurement data seen in [4] where the rate of increase is more consistent and slows down as time progresses. There is a rather quick change of temperature increase around 40 seconds and at approximately 110 seconds, the rate of change seems to increase. These characteristics cannot be seen in [4] and Fig. 8. The analytical solution to the bio-heat equation was fitted to the data acquired from measurements. Just as with the simulation, it was done using a non-linear least squares method. The results can be seen in

Fig. 12. The estimation of the electromagnetic properties of the phantom was made in the same manner as in the previous subsection and the results can be seen in Table IV. The value for the complex permittivity deviates from the expected value, which should be near the complex permittivity of water. This is due to the fact that the phantom consists mostly of water (97%). The function generator was set to 4.705 GHz, thus the expected skin depth should be slightly larger than the simulated value of 8.17 mm. The expected complex permittivity of the phantom should be close to $74 - j20$ due to its high water content but given the expected value of the skin depth, it will likely be slightly less than $74 - j20$. This is due to eventual variations in ε_s to match the skin depth, resulting in different values for the complex permittivity. The value for the complex permittivity of water can be seen in [9] and values for the relaxation time of water in the frequency range 0 - 30 GHz at different temperatures can be seen in [8]. The estimated values from the fitting procedure is given by $q = 4.3706 \times 10^5$ and $L = 0.0033$ mm. This is not necessarily a bad fit and the 95% confidence intervals for q and L are respectively given as $(4.1045 \times 10^5, 4.6361 \times 10^5)$ and $(0.0029127, 0.0037039)$.

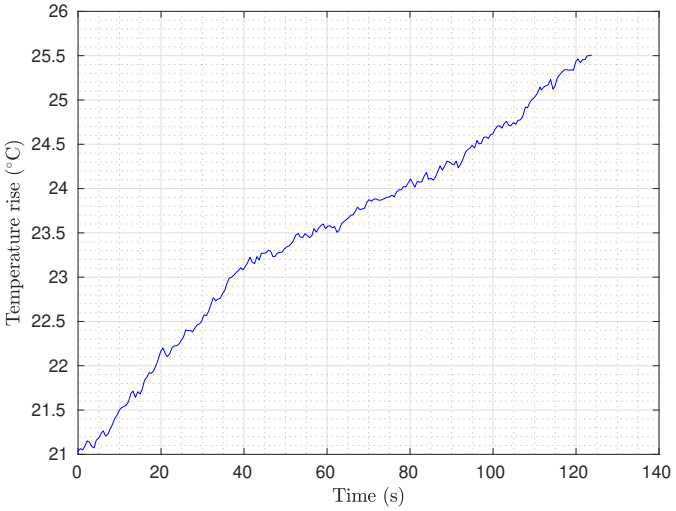


Fig. 11. Data of temperature increase on the surface of the phantom acquired from experimental measurements.

TABLE IV
ESTIMATED ELECTROMAGNETIC PROPERTIES OF MANUFACTURED PHANTOM

δ, mm	I, W/m^2	$\Delta\epsilon$	τ, ps	ϵ^*
6.6	953.89	145.9	9.1	$141.16 - j36.6$

Moreover, the mean squared error is 0.0196 and the average value of the residuals is -0.0044 which indicates a slight undershoot of the estimated parameters.

V. DISCUSSION

The aforementioned results were divided into two parts, simulation and measurements. The measurements are with

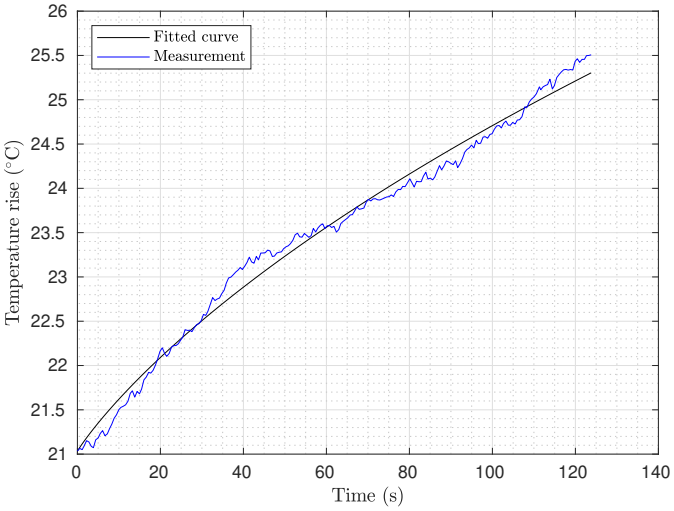


Fig. 12. The solid line shows the analytical solution to the bio-heat equation fitted to the data from the experimental measurements.

reference to the experimental results. The estimated complex permittivity with respect to simulations done in CST correspond well with the expected electromagnetic properties of the constructed material in CST. As was mentioned earlier, the complex permittivity of the material in CST was defined through the $\tan(\delta_e)$, thus establishing a reference point for ϵ^*. The heat transfer coefficient for the material in CST was defined in section III. It was described as a convective heat transfer coefficient. Its emissivity was also defined which gives an indication of thermal loss through radiation.

Moreover, the heat transfer coefficient in the analytical solution to the bio-heat equation is with reference to the total heat loss due to radiation, convection and evaporation. Given the temperatures involved, it was assumed that evaporation did not have an impact to the total heat loss. The heat transfer coefficient in CST does not take the total heat loss into account as is done in the analytical solution to the bio-heat equation. Thus, used in the fitting procedure was modified slightly to include the effects of radiation to heat loss. The amount added is given by equation (7) where T, in this setting, is thought of as the average surface temperature over the defined period of time. Taking this observation into account, the results from the simulation in CST are in good agreement with the complex permittivity of water at 5 GHz. The results from experimental measurements differ considerably from the expected complex permittivity which should resemble the value for water, due to the fact that the phantom mainly consists of water. There is a distinction that has to be made as it relates to the results from the experimental measurements. On one hand, the reliability of the acquired data can be discussed but so can the validity of the Debye-model used in the framework of this project. The thermal properties of the manufactured phantom was not determined, this can lead to a lot of uncertainty regarding the actual values. Nevertheless, it was assumed during the fitting procedure that the thermal properties of the phantom resembled the values found in [9]. Varying the thermal properties in the analytical solution to the bio-heat equation during the fitting

process does not have a large impact on the resulting value for the skin depth. It does however impact the resulting complex permittivity, as the Debye-model seems to be quite sensitive to changes in the skin depth. The estimated skin depth of 6.6 mm differ by 1.4 mm from the expected skin depth of approximately 8 mm, this corresponds to a difference of about 18%. A skin depth of 6.6 mm results in a complex permittivity that can be seen in Table IV. If the skin depth changes from 6.6 mm to 8 mm with the same value for the relaxation time τ but with a modified ϵ_s, the complex permittivity becomes approximately $95 - j25$. Thus a change of 1.4 mm in skin depth results in a difference of approximately 33 % to the real part and approximately 32 % to the imaginary part. Thus, smaller variations in skin depth can result in larger variations in the complex permittivity obtained from the Debye-model. This also shows the importance of reducing the uncertainties of data acquired through experimental measurements. The aforementioned comparison is simply an observation, a potential source of error in the comparison is that the frequency is set to the same value when comparing the same skin depth. The relaxation time of water is with reference to [8] and it is more reasonable to use the same value for the relaxation time as the values are given in a frequency range. The application of the Debye-model resulted in expected values for the complex permittivity in a simulation setting and the model was used in [4] with resulting values that were in good agreement to measurements by [4] done with co-axial cables. Thus, it can be said that the application of the Debye-model is validated. The potential problem arises in applying the method with reference to the data acquired in Fig. 11. During the fitting procedure, a model function has to be specified. The model function is the analytical solution to the bio-heat equation. Thus, not all forms of data will correlate well to the application of the model function in the non-linear least squares method. This is not to say that the result of the fitting procedure yields an incorrect fit but rather that the measurement data seen in Fig. 11 is not entirely applicable with respect to the model function used. The characteristics of the expected curve can be seen in [4]. The characteristics of the curve in Fig. 11 was discussed in section IV. Thus, it does not necessarily seem reasonable to reduce the time span for the fitting procedure as the choice of different time spans would impact the resulting skin depth and thus the complex permittivity. Given the characteristics of the curve, the choice of time span would not be certain with regards to the estimation of properties.

A. Future work

Whilst there are a number of uncertainties regarding the experimental results, certain improvements can be made. It could potentially be of interest to manufacture phantoms of different materials and to apply the same method to estimate the complex permittivity, thus providing more information regarding the choice of material. Moreover, the experimental setup as seen in Fig. 1 and Fig. 2 could be modified such that there is less equipment around the phantom. A potential source of error could have been the usage of the IR-camera and a potential improvement can be achieved by changing its settings in terms of for example sampling rate. Moving the setup to different spots in the anechoic chamber to analyze and get a better understanding of how the specific placements affect temperature measurements can be done in accordance with creating a larger distance of separation between the phantom and the measurements setup. The emissivity was set to 0.96, this value is rather high which reduces measurement errors as this allows for the reflected temperature to be set to ambient (20°C). Moreover, creating a larger distance between the phantom with the IR-camera and the measurement setup could result in a better understanding of potential convective effects resulting from output of heat from for instance the function generator. The IR-camera was situated fairly close to the phantom, thus taking the amount of pixels into consideration and thereby reducing measurement errors related to distances. Whilst the absorbed power density which contains the power reflection coefficient R is estimated through the fitting procedure, thus taking reflections into consideration, varying the intensity of the incident electromagnetic wave and studying its effects could be useful in gaining more understanding of the thermal processes involved and the characteristics of the data acquired.

VI. Conclusion

The project involved estimating the complex permittivity of a phantom. This was done in a simulation setting in CST and experimental measurements were also taken. The uncertanties regarding data of surface temperature increase in CST is minimized and the resulting complex permittivity is in good agreement with the constructed phantom in CST, thus effectively showing that the method works. The uncertanties related to the experimental measurements can be seen in the corresponding results where the estimated complex permittivity is not in good agreement with the expected complex permittivity, this was further discussed in section V. However, given the uncertainties of the experimental measurements, it is not valid to claim that the method does not work. It provides an alternative way to measure and estimate the complex permittivity of materials which can be used to verify safety guidelines for implementation of new technology.

Acknowledgment

The authors would like to thank the supervisors Oscar Quevedo-Teruel, Oskar Zetterström and Freysteinn Vidarsson for their academic guidance.

References

[1] (2021, Apr.) Ericsson mobility report november 2020. [Online]. Available: https://www.ericsson.com/4adc87/assets/local/mobility-report/documents/2020/november-2020-ericsson-mobility-report.pdf

[2] *C95.1-2019 - IEEE Standard for Safety Levels with Respect to Human Exposure to Electric, Magnetic, and Electromagnetic Fields, 0 Hz to 300 GHz*. IEEE, Nov. 2019.

[3] S. Gabriel, R. W. Lau, and C. Gabriel, "The dielectric properties of biological tissues: Iii. parametric models for the dielectric spectrum of tissues." *Physics in medicine and biology*, vol. 41, pp. 2271–2293, Apr. 1996.

[4] N. Chahat, M. Zhadobov, R. Sauleau, and S. I. Alekseev, "New method for determining dielectric properties of skin and phantoms at millimeter waves based on heating kinetics," *IEEE Transactions on Microwave Theory and Techniques*, vol. 60, pp. 827–832, Mar. 2012.

[5] O. P. Gandhi and A. Riazi, "Absorption of millimeter waves by human beings and its biological implications," *IEEE Transactions on Microwave Theory and Techniques*, vol. 34, pp. 228–235, Jan. 1986.
[6] C. Alabaster, "Permittivity of human skin in millimetre wave band," *Electronics Letters*, vol. 39, pp. 1521 – 1522, Nov. 2003.
[7] S. Alekseev and M. Ziskin, "Human skin permittivity determined by millimeter wave reflection measurements," *Bioelectromagnetics*, vol. 28, pp. 331–339, Jul. 2007.
[8] W. J. Ellison, "Permittivity of pure water, at standard atmospheric pressure, over the frequency range 0 – 25 thz and the temperature range 0 – 100 ° c," *Journal of physical and chemical reference data*, vol. 36, pp. 1–18, Mar. 2007.
[9] A. K. Fall, P. Besnier, C. Lemoine, M. Zhadobov, and R. Sauleau, "Experimental dosimetry in a mode-stirred reverberation chamber in the 60-ghz band," *IEEE transactions on electromagnetic compatibility*, vol. 58, no. 4, pp. 981–992, Apr. 2016.
[10] S. J. Orfanidis. (2019, Aug.) Uniform plane waves. [Online]. Available: https://www.ece.rutgers.edu/~orfanidi/ewa/ch02.pdf
[11] D. K. Cheng, *Field and wave electromagnetics*, 2nd ed. Harlow, Essex, England: Pearson, 2014, pp. 367–370.
[12] *Thermal and Mechanichal Simulation*, 2020th ed., Dassault Systèmes, Paris, France, Aug. 2019.

Simulation and Measurement of Body Absorption for 5G/6G Frequency Bands

Frida Skarf and Maximilian Dahlenmark

***Abstract*—For the future implementation of high speed communication, safety remains one of the main concerns. To ensure the safety of new applications, specifically the new 5G antennas, it is crucial to know that they will not cause any harm to the human body. There are a few ways to test how safe a system using high frequency radiation is but the industry standard is by using the Specific Absorption Rate (SAR). The SAR is directly correlated to the initial temperature rise in the volume exposed to radiation which is what the method used in this report is based on. The temperature rise in a skin-like phantom due to 5 GHz exposure was recorded using an IR-camera, which in turn was used to calculate the SAR. The purpose of this report was to test if this method is a valid way of obtaining the peak surface SAR. It was concluded that the method is valid but there are some uncertainties in regards to abstracting the method to far-field exposure for our considered frequency. The SAR value that is achieved in this report is 333.4 W/Kg which is high in relation to the SAR-limits in IEEE guidelines, although the set up is not supposed to reflect a realistic use of the antenna. This is due to the fact that the waveguide in the setup is close to measurement sample, and has a higher intensity than is to be expected from real world applications. The method may be applicable for far-field exposure with a higher frequency as that would concentrate the measurable heat to the surface of the measurement sample and would also carry more energy by default.**

***Sammanfattning*—För all framtida implementering av höghastighetskommunikation kommer säkerhetsgarantin vara en av de stora frågorna. För att fastställa säkerheten hos nya tillämpningar, specifikt 5G antenner, är det viktigt att veta att det är oskadligt för människor. Det finns olika metoder för att påvisa säkerheten i applikationer som använder sig av högfrekvent strålning men industristandarden är att använda SAR (Specific Absorption Rate). SAR är direkt korrelerat med värmeutvecklingen i volymen utsatt för strålning vilket är det faktum denna rapport är baserad på. Temperaturökningen i en skinnliknande fantom från 5 GHz-strålning mättes med en IR-kamera. Detta användes för att räkna ut SAR-värdet. Poängen med rapporten var att testa om denna metod är acceptabel för att räkna ut högsta SAR-värdet i en punkt. Slutsatsen är att denna metod fungerar men det kan finnas problem med att abstrahera metoden till fjärrfält för vår tilltänka frekvens. Det uppnådda SAR-värdet är 333.4 W/Kg vilket är högt jämfört med IEEE-standarden, detta är på grund av att uppställningen inte är tänkt att återspegla en realistisk situation. Antennen är för nära fantomen och intensiteten är också högre än vad som används i vanliga fall. Metoden kan möjligtvis användas vid fjärrfältstrålning vid högre frekvenser i och med att detta leder till en värmeökning koncentrad vid ytan. Dessutom har högre frekvenser högre energi vilket innebär en större värmeökning.**

***Index Terms*—SAR, 5G, antennas, anechoic chamber, skin-like phantom, bio-heat equation**

Supervisors: Oskar Zetterström, Freysteinn Viðar Viðarsson, Oscar Quevedo Teruel

TRITA number: TRITA-EECS-EX-2021:175

I. Introduction

The world has a greater demand of data than ever before and therefor a demand for high speed communication of said data. In addition more devices are connected to the internet and the usage continues to grow. As such the new solution for wireless communication must be able to accommodate the ever-growing user base [1].

One solution is moving up in frequencies to increase both bandwidth and speed, and these higher frequencies are in part what the fifth generation of cellular networks '5G' uses. With a data traffic increase of about 60% per year due to new connected services and more videos streamed, a speed and capacity increase is especially important [1]. 5G's purpose is to increase the capacity for social media, video streaming and other things we are already doing today, but also for new innovative use cases such as securely streaming high-quality video from an ambulance to the hospital and enabling a range of new types of smart devices (like self-drivning cars) and industry digitalization [1].

The implementation will provide numerous challenges for the engineers that are developing the systems, but also some potential problems with safety in relation to human exposure of said systems. Higher frequencies can heat biological tissue and increase body temperature, which some fear might cause health complications. Unlike claims that 5G would cause COVID-19 [2] there are some legitimate concerns with using high energy radiation as the source of all communication.

To concretize, the high frequency has the ability to penetrate the human skin, this will result in a small increase in temperature. Just how much energy is absorbed is called the Specific Absorption Rate (SAR). We want to calculate SAR for the human body to determine how much heat is developed from the same waves used by 5G. SAR is essentially a measurement of how much energy different materials absorb from waves of a certain frequency, and there is a limit to what is considered safe for humans [3]. We have looked into the question of how SAR can and should be measured. One proposed method is to radiate a small phantom of skin-like material with a 5G antenna and measure the increase in temperature [4], [5]. SAR has a direct correspondence to the initial temperature rise so one should be able to calculate the SAR for the small sample [6]. A phantom, in this case, is a replacement of human tissue and can be made up of several solutions as long as they mimic the electrothermal conditions of human skin [7], [8]. Using an approximate solution to the bio-heat equation [7] with slightly different parameters allows us to match the lab data to a theoretical model which in turn can be used to calculate SAR.

The experimental set up for the temperature measurements is inspired by Fall et. al. [5]. The main difference being the use of the anechoic chamber instead of the reverberation chamber as well as radiating with a lower frequency. The anechoic chamber allows for a more precise conclusions to be drawn since we do not have to consider second hand radiation. Some of the data from the reports mentioned have been used to simulate and test the set up. The actual set up consists of the phantom in an anechoic chamber with the antenna and an IR-camera directed at the phantom. This method was the one tested in this report.

II. Electrothermal model

Determining the SAR in the skin-like phantom starts with constructing a theoretical model that replicates the rise in temperature on the surface of the phantom due to radiation. We claim that the temperature rise will follow the solution to the bio-heat transfer equation with one-sided radiation. One-sided radiation means that the sample is radiated with a beam directly aimed at the surface and that no energy absorption will happen at the sides or bottom of the sample. This is a result of the measurements being performed in an anechoic chamber, i.e. no second hand radiation. The first step is therefore to solve the bio-heat equation.

A. Solution to the bio-heat transfer partial differential equation

The aim is to theoretically evaluate the temperature rise on a surface of a sample exposed in an anechoic chamber. This is usually a 3-D problem where the 3-D heat equation would have to be solved. However, our interests lie in the temperature far away from the sample edges, in the middle, which is why solving the 1-D heat equation will suffice [5]. The 1-D PDE will be solved analytically. To emulate the conditions in the anechoic chamber the following PDE will be solved:

$$\frac{1}{\alpha_t}\frac{\partial T_s}{\partial t} = \frac{\partial^2 T_s}{\partial z^2} + \frac{Q(z,t)}{k_t} \tag{1}$$

where T_s [°C] is the temperature of the sample. We will now consider the relative temperature $T = T_s - T_a$ where T_a [°C] is the ambient temperature and solve it under the following conditions:

$$T(\infty, t) = 0, \qquad t \geq 0 \tag{2}$$

$$\frac{\partial T(0,t)}{\partial z} = \zeta T, \qquad t \geq 0 \tag{3}$$

$$T(z,0) = 0 \tag{4}$$

t [s] is time, z [m] is the depth coordinate, and $Q(z,t)$ [W/m^3] is the heat deposited into the phantom. The constants are the thermal diffusivity $\alpha_t = k_t/(\rho \cdot C)$ [m^2/s], where k_t $[W/(m{\cdot}K)]$ is the thermal conductivity, ρ $[kg/m^3]$ the density and C $[J/(kg{\cdot}K)]$ the specific heat capacity, $\zeta = h/k_t$ [m^{-1}] is the the thermal effusivity and h $[W/(m^2 \cdot K)]$ the heat transfer coefficient. $Q(z,t)$ is defined as

$$Q(z,t) = \frac{2(1-R)}{\delta} I_0 e^{-2z/\delta} H(t) \tag{5}$$

where δ [m] is the skin depth, R is the power reflection coefficient, I_0 [W/m^2] is the average input power and $H(t)$ the Heaviside function [6].

From PDE solution methods the transient solution for the temperature rise on the surface (i.e., at $z = 0$) of the phantom is given by

$$T(t) = qL^2\Bigg\{2\frac{1/L+\zeta}{\zeta} - 1 + e^{\alpha_t t/L^2}\operatorname{erfc}(\sqrt{\alpha_t t}/L) \left[\frac{1}{2} + \frac{1/L+\zeta}{2(1/L-\zeta)}\right] - \frac{\zeta/L^2}{\zeta^2/L - \zeta^3} e^{\zeta^2\alpha_t t} \operatorname{erfc}(\zeta\sqrt{\alpha_t t})\Bigg\} \tag{6}$$

where $q = 2(1-R)I_0/(\delta \cdot k_t)$ and $L = \delta/2$ [7].

B. SAR calculation using the bio-heat solution

The specific absorption rate is defined as the power absorbed per unit mass and can be related to the internal electric field $|E|$, the conductivity σ, and the mass density of the sample ρ:

$$\text{SAR} = \frac{\sigma |E|^2}{\rho} \tag{7}$$

The SAR of electromagnetic energy in a sample is also proportional to the initial temperature rise rate and can be determined as follows:

$$\text{SAR} = C\frac{dT}{dt}\Big|_{t=0} \tag{8}$$

where C is the specific heat of the sample being radiated. This assumes that measurements are made under "ideal" non-thermodynamic circumstances, i.e. no heat loss by thermal diffusion, heat radiation, or thermoregulation (blood flow, sweating, etc.) [4]. Since measurements will be performed on a phantom the model is applicable. The definition involving initial temperature rise is useful since the analytic temperature as a function time has previously been obtained. By using the previous result in Eq (6) and differentiating the function with respect to t the following derivative is obtained:

$$\frac{dT}{dt} = \frac{q\alpha_t}{\zeta L - 1}\Big(\zeta L e^{\zeta^2\alpha_t t}\operatorname{erfc}\zeta\sqrt{\alpha_t t} - e^{\alpha_t t/L^2}\operatorname{erfc}(\sqrt{\alpha_t t}/L)\Big) \tag{9}$$

evaluated in $t = 0$ is just

$$\frac{dT}{dt}\Big|_{t=0} = q\alpha_t = \frac{2(1-R)I_0}{\delta\rho C} \tag{10}$$

According to Eq (10) the only thing that determines the SAR is the skin depth δ and the deposited power $I = (1-R)I_0$:

$$\text{SAR} = \frac{2I}{\delta\rho} \tag{11}$$

III. Method

The aim is to perform temperature measurements on a skin-like phantom being exposed to 5 GHz radiation and then using the resulting temperature curve to calculate the SAR. As previously realised, I and δ is needed to determine the SAR. The easiest way to obtain them is fitting the analytical solution (6) to lab results with respect to these variables. The fitting will be done using a non-linear least square method.

A. Model verification

The electromagnetic FEM-software CST Studio Suite is used to perform simulations similar to the one in the lab. CST has the ability to calculate the temperature rise in a sample exposed to electromagnetic radiation. In addition CST can also directly calculate SAR. The idea is to verify the method by comparing the directly calculated SAR to the one obtained by fitting the bio-heat solution to the generated temperature curve. A model of a slab of water in air is exposed to radiation with a frequency of 5 GHz. A SAR- and temperature simulation is performed on the slab under the same exact conditions. The simulation model is shown in Figure 1.

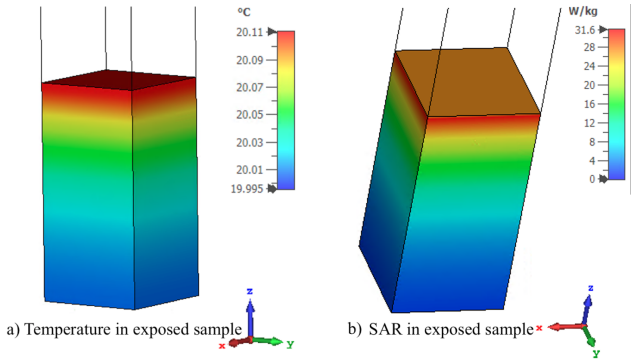


Figure 1. Temperature and SAR distribution in sample from E = 1000 V/m.

The boundary conditions are chosen to replicate conditions in the lab, Figure 2.

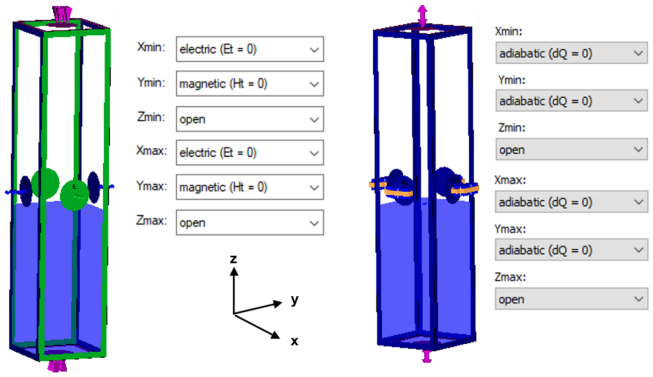


Figure 2. Electromagnetic and thermal boundary conditions.

The z-bounds are open, which means the z boundary is extended to infinity. The adiabatic conditions (right in Fig 1) means we have no boundary losses. Since the wavelength is small in comparison to the phantom the boundary can be set this way. The bottom of the sample is largely unaffected by the radiation as can be seen in Figure 1 which suggests that the infinite model is both practical and appropriate to emulate lab conditions. From the simulations the SAR at the center of the surface is noted. The SAR is recorded 1 mm below the surface to avoid problems with discretization in the model. This phenomenon can be observed in Figure 1.b where the SAR value on the top surface is mismatched with the rest of the gradient which can be seen from the side. The temperature rise on a small part of the same surface produces the needed temperature curve. The SAR is calculated using both methods with the following results, see Table I, where SAR curve fit is the method intended to be used with the actual measurements done in the anechoic chamber. To illustrate, the curve fit for $E = 1500$ V/m is shown in Figure 3.

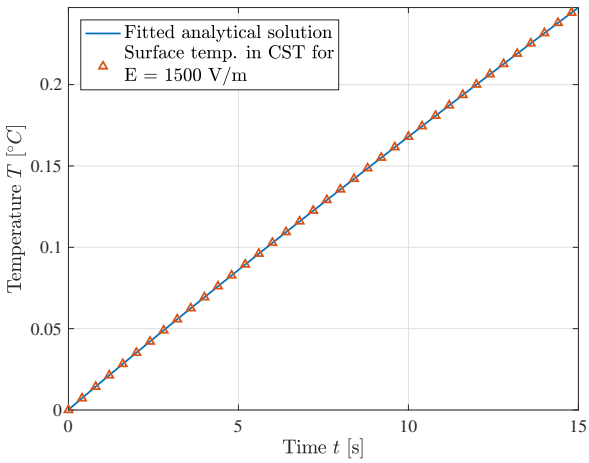


Figure 3. Surface temperature plot for $E = 1500$ V/m fitted with the analytical solution, Eq (6).

Table I
SAR-calculations from CST simulation

E-field [V/m]	SAR CST *on* surface [W/kg]	SAR CST 1 mm *under* surface [W/kg]	SAR curve fit [W/kg]
1000	26.81	30.24	30.29
1500	60.33	68.04	68.60
2000	107.3	121.0	122.4

By varying the electric field different SAR values are obtained. Good agreement between the values obtained from CST and the values from the curve fitting method was found, as can be observed the error margin is very small for the SAR 1 mm below the surface.

B. Setup for temperature measurements

The idea is to perform the measurements in an anechoic chamber. As can be seen in Figure 4 the IR-camera (FLIR A65SC) is aimed towards the phantom at an angle and the antenna is suspended directly over the phantom. The antenna is fed with a signal generator at 5 GHz.

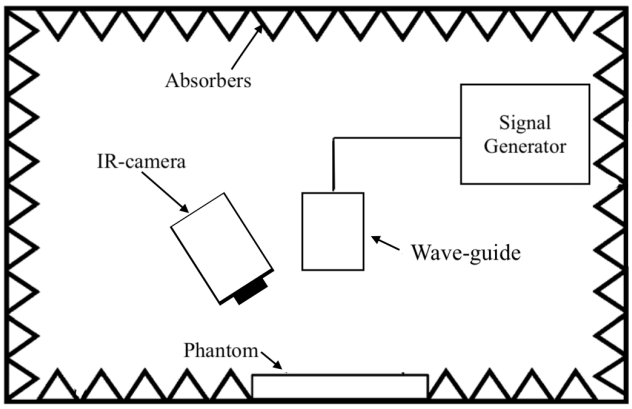


Figure 4. Schematic representation of lab setup (not to scale).

1) Phantom: The phantom was produced one day in advance. Its composition is mostly water, with gelatin to solidify it. It measures 11x11x3 (base x width x height) cm, see Figure 5. Due to the high water concentration, the permittivity of the phantom is close to the permittivity of water, which in turn is relatively close to skin.

2) Equipment: To record the temperature rise we used a FLIR A65SC IR-camera with accuracy of $\pm 5\%$ and a thermal sensitivity of $<$50 mK. Important to note is that since SAR is only reliant on dT/dt the accuracy of the camera will not have a great impact. We used the HP 8665B signal generator to excite a waveguide.

3) Anechoic chamber: The measurements was performed in the anechoic chamber at KTH. The purpose of an anechoic chamber is to completely absorb reflections of the electromagnetic waves. This exposure is most similar to the future real world applications of 5G antennas. The new antennas will most likely be directional ones as opposed to the currently omnidirectional ones, which means that the direct radiation conditions in the anechoic chamber is more suited for this experiment.

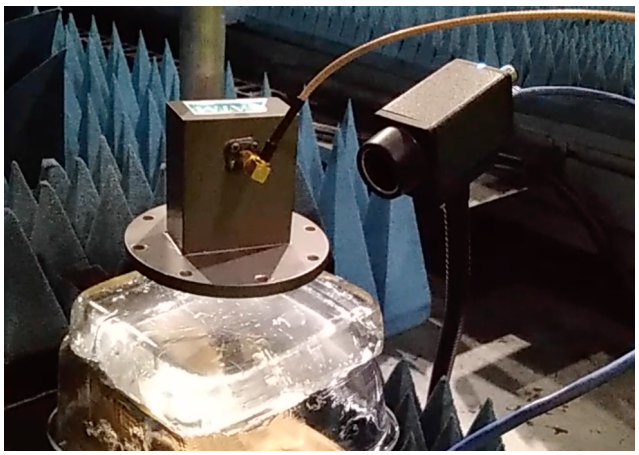


Figure 5. Picture of lab setup with IR-camera, waveguide and gelatin phantom.

IV. Results

The phantom experienced a significant heating over the 90 second time period of illumination, Figure 6.

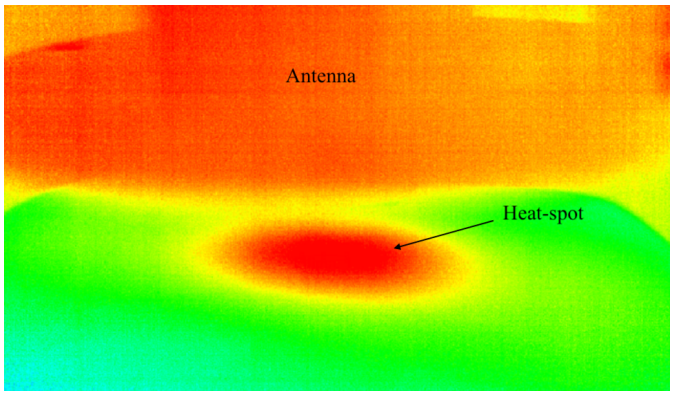


Figure 6. Heat map for lab measurements.

By fitting the deposited power $I = (1-R)I_0$ and the skin-depth δ in the bio-heat solution (6) the SAR can be determined. The constants in table II were used for the fit.

Table II
PHYSICAL PROPERTIES OF THE PHANTOM

Thermal conductivity k_t [W/(m·K)]	0.66
Density ρ [kg/m^3]	1000
Specific heat capacity C [J/(kg·K)]	3770
Heat-transfer coefficient h [W/(m^2·K)]	8.16 [5]

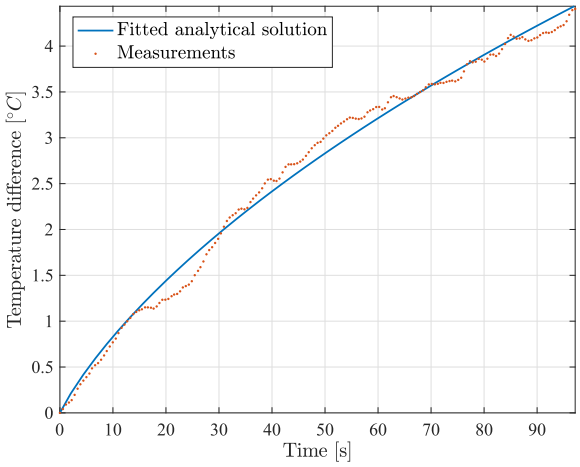


Figure 7. Complete measurement series with analytical solution fitted to entire 90 s illumination period.

Both the first 10 s (figure 8) and the entire heating process (figure 7) was fitted with a theoretical solution, although it is known that for these purposes the 10 s fit is more relevant since the SAR is dependent on the temperature rise at $t = 0$. This fit gave a peak surface SAR of 333.4 W/kg.

Some minor disturbance could be observed in the measured data, otherwise the temperature rise behaved as expected.

The fitting procedure resulted in the SAR-values presented in Table III.

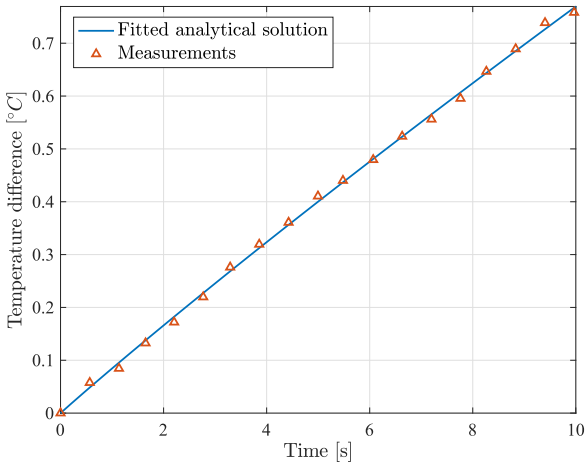


Figure 8. Partial measurement series with analytical solution fitted to the first 10 s of illumination.

Table III
FITTED PARAMETERS WITH SAR

	δ [mm]	I [W/m^2]	SAR [W/kg]
90s fit (fig 7)	4.066	974.4	544.6
10s fit (fig 8)	14.97	2495	333.4

V. DISCUSSION

A clear rise in temperature from the exposure was observed and the temperature rise in the phantom over a time period could be measured. The initial temperature rise seemed reasonable in relation to previous studies. The heating was centred and no unexpected gradients were observed, see figure 6. The resulting skin-depth δ was also in agreement with the theoretical value for the considered frequency and material, see Table III.

The aforementioned disturbances we suspect is a result of the IR camera sensitivity. Another uncertainty is the time between starting the temperature recording and turning on the signal generator. This time has been observed to be about 5 s. Therefore, at the very beginning of the measurement series, we cannot expect to fit the analytical solution. As such, we performed the fitting process in the 5–15 s and 5-95 s range instead of 0–10 s and 0-90 s respectively.

From this we could determine the peak surface SAR. The SAR-values obtained in Table III, namely the 10 s fit SAR-value of 333.4 W/kg, is reasonable considering the measurement set up, with the waveguide close to the sample and the high intensity radiation. It is also in agreement with the results in [5] where a peak surface SAR in the range of 333-397 W/kg was recorded with a similar lab setup.

For the purpose of determining the peak surface SAR of a sample exposed to radiation the method is viable. This is verified by the simulation done in section III)A as it shows that the SAR value that is achieved from the simulation is very close to the SAR calculated with our method, as long as the heating is measured correctly. The method's ability to be applied to temperature measurements in a lab setting was also confirmed.

The use of a water-gelatin mix phantom to emulate human skin is valid due to skin consisting of mostly water with collagen, and collagen is found in gelatin. For future applications we would consider this phantom a sound choice, although it would benefit the method to do more of a thorough analysis of the proportions of water to gelatin. In addition could for example both salt and sugar be added to the phantom to more closely resemble human skin.

There are some potential problems with this method. The determination of SAR is easier for near-field exposure, as opposed to far-field exposure, since the internal fields are confined primarily to the volume directly adjacent to the exposure. As a result an abstraction of the method to far field case will most likely prove difficult for the considered frequency. Complications with the near-field case arise when the whole body of a person or animal is exposed to high frequency radiation, which by reflection and standing waves can cause localized heating. The resulting intense, local "hot spots" are due to resonant conditions existing in these regions. As such the local SAR may exceed the average whole-body SAR by factors exceeding 100 times. The local SAR values and the SAR distribution in biological objects cannot be measured without producing relatively large measurement uncertainties, regardless of the instrumentation used. Under ideal plane-wave exposure conditions, the maximum local (point) SAR can be 20 to 100 times greater than the whole-body-averaged SAR [4].

The equipment also has limitations, the IR-camera's performance decays with lowering intensities (i.e lower temperature differences) which means that this method might not give reliable measurements at "safer" intensities. Safer in this sense are in reference to what would be expected in the real world and more in line with guidelines set by IEEE [3]. The method measures *point-SAR* which is not used as a safety standard measurement.

Although a move up to higher frequencies might solve some of these issues. At 5 GHz the waves penetrate deeper into the phantom which is problematic since the heating will spread from the surface. By increasing the frequencies to ones where the skin depth is less than a millimetre, the heating is mostly limited to the surface which enables easier recording of temperature rise.

The upsides of the method is that it is simpler to use when frequencies are high, the other methods of measuring SAR have proven difficult to realise with these higher frequencies since the penetration depth is very small. Measuring the total temperature rise in a larger body would not give a satisfactory result when the skin depth reaches sufficiently small values. In addition, the equipment is easily accessible, mainly a functioning IR-camera and a signal generator.

VI. CONCLUSION

We conclude that the method to determine SAR was successful and can be used. The measured data was in accordance to a theoretical solution and parameters could be fitted to it. This can be supported with the simulations that showed the similarity between the SAR calculated using the method in this

report and SAR calculated by CST Studio Suite. Although no conclusions could be drawn from the measured data in regards to far-field exposure SAR averaged over the whole body. An extension of the project is therefore to find a way to perform the same measurements with greater distances from antenna to sample. This would be a setup more similar to real world applications.

ACKNOWLEDGMENT

The authors would like to thank Oskar and Freysteinn for all their valuable advice, their knowledge and reliability. For always taking the time to assist and offer solutions. You have been of great help throughout the entirety of the project and we of course could not have done it without you. Also a great thanks to Oscar who with his always positive attitude inspired us to make our project the best it could be.

REFERENCES

[1] Ericsson. (2021, April) 5g vs 4g: What is the difference? Stockholm. [Online]. Available: https://www.ericsson.com/en/5g/what-is-5g/5g-vs-4g

[2] A. Satariano and D. Alba. (2020, April) Burning cell towers, out of baseless fear they spread the virus. New York. [Online]. Available: https://www.nytimes.com/2020/04/10/technology/coronavirus-5g-uk.html?referringSource=articleShare

[3] "Ieee standard for safety levels with respect to human exposure to electric, magnetic, and electromagnetic fields, 0 hz to 300 ghz," *IEEE Std C95.1-2019 (Revision of IEEE Std C95.1-2005/ Incorporates IEEE Std C95.1-2019/Cor 1-2019)*, pp. 1–312, 2019.

[4] "Ieee recommended practice for measurements and computations of radio frequency electromagnetic fields with respect to human exposure to such fields, 100 khz-300 ghz," *IEEE Std C95.3-2002 (Revision of IEEE Std C95.3-1991)*, pp. 1–126, 2003.

[5] A. K. Fall, P. Besnier, C. Lemoine, M. Zhadobov, and R. Sauleau, "Experimental dosimetry in a mode-stirred reverberation chamber in the 60-ghz band," *IEEE Transactions on Electromagnetic Compatibility*, vol. 58, no. 4, pp. 981–992, Aug 2016.

[6] S. Alekseev and M. Ziskin, "Local heating of human skin by millimeter waves: A kinetics study," *Bioelectromagnetics*, vol. 24, no. 8, pp. 571–581, 2003. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/bem.10137

[7] N. Chahat, M. Zhadobov, R. Sauleau, and S. I. Alekseev, "New method for determining dielectric properties of skin and phantoms at millimeter waves based on heating kinetics," *IEEE Transactions on Microwave Theory and Techniques*, vol. 60, no. 3, pp. 827–832, 2012.

[8] S. Gabriel, R. W. Lau, and C. Gabriel, "The dielectric properties of biological tissues: III. parametric models for the dielectric spectrum of tissues," *Physics in Medicine and Biology*, vol. 41, no. 11, pp. 2271–2293, Nov 1996. [Online]. Available: https://doi.org/10.1088/0031-9155/41/11/003

CONTEXT K
ELECTROTECHNICAL MULTIPHYSICS SIMULATION

POPULAR DESCRIPTION

Building a Digital World to Optimize our own

Imagine if we could prevent disasters by knowing beforehand that they would happen. Imagine that wild ideas could be tested and perfected with just a push of a button. How would the world then change? No more trial and error. This is the future that simulations could provide us with.

Computer simulation allows us to test scenarios with difficult set ups and in a way predict the future. These methods could be used to develop products to make your day-to-day life easier or to minimize negative impact on the environment. The nose of an airplane could for example be optimized to decrease air resistance, which in turn reduces fuel consumption. A model like this one would need to take several fields of physics into account.

Computer simulations grant freedom and flexibility to test even the most extreme scenarios. If you were to physically test a model of a boat you would have to recast the hull for every profile and dimension. Were you instead to use a computer model you could test multiple profiles and geometries in a smaller time frame and using less resources. You could do all that and test for different materials all in one go. A physical model would only be required in the final stage of testing. A problem that could arise with larger, more complex models is the need for processing power. Many of us do not have endless time, and running a complex model on a weak computer would take too long to be practical for anyone.

A computer simulation creates a virtual version of the product or system. This digital twin can be used to simulate the multiple physical phenomena simultaneously acting on the system. These simulations are possible thanks to powerful processors, software and engineers trained in multiphysics modeling. You can create a whole virtual world where the only limitation is your imagination, and access to computational power of course.

SUMMARY OF PROJECT RESULTS

The cornerstones of multiphysics simulation are the software making it possible and the physics it is based on. The software used during the project in this context has been COMSOL Multiphysics, which uses the finite element method to model and simulate problems. An important aspect in creating and using a model is to simplify it and set constraints. Many advanced and complex models demand a supercomputer to run fast enough to be practical but even a supercomputer can have a hard time with these models. To bypass this limitation, simplifications to the physics equations used need to be made. These can for example be to assume that a value is constant or to use a relation that in reality is only valid for a certain span of parameters.

The focus of project group K1 has been on simulating the effect on airflow given by plasma actuators. Plasma actuators are devices that create plasma between two electrodes and depend on different parameters. These parameters are influenced by the properties of the applied current and the geometry of the actuator. The idea is to use plasma actuators to reduce the air resistance on trucks, aircraft and other vehicles and therefore reduce the fuel consumption. Previous physical trails in this area have been numerous. The challenge during these trials has proved to be that simulating the physics of plasma actuation is too time consuming. For this reason, simplifications have had to be done. The project group has studied these simplified models and then used one of them to perform several parametric analysis of plasma actuation. The goal has been to optimize the balance

between fuel consumption and the energy necessary to power the actuator i.e. maximize the active power compared to the total power input.

IMPACT ON SOCIETY AND ENVIRONMENT

In many ways, multiphysics simulations have a positive environmental impact in that a lot of resources can be saved, which makes simulations more sustainable than other kinds of testing. Computers, power and other aids that are needed for running simulations are also resources, but compared to many other resources these are not the ones with the biggest environmental impact when being manufactured. Computer modelling also saves money, as there is no need for active factories during the testing period. This eliminates the need for resources other than the necessary processing power and the people working with the simulations.

There are also cases where the simulations can be used for research, especially when it comes to predictions of environmental changes. Models and simulations can also be useful when developing various tools and devices for decreasing society's negative environmental impact. For example, the plasma actuator studied in project K1 is being developed and optimized to decrease the air resistance on trucks and therefore save fuel and reduce emissions.

Something to reflect on is the ethics of computer simulation. One ethical aspect to consider is transparency. When using a product that is mainly run by a computer, the user or customer is mostly unaware of what went into the testing of the system. It is the developer's responsibility to make the customers aware of any and all limitations and faults in a system, so that they know how to use the system in the best way. As the developer, it is your responsibility to inform the customer of what physical phenomena you have tested, so that the system or product will not be put under stresses that were not accounted for in simulations. It also goes hand-in-hand with being transparent about your own limitations and abilities with multiphysics simulations. In some cases, a fault in the system or misuse of it could cause someone to lose their life. For example in the space industry, you can often not test every system before launch and so the astronauts put their lives in the hands of the simulation engineers.

Another ethical aspect is whether your model is accurate enough to represent reality. The model is often based on or legitimized by experimental data, but exactly how close it is to real life is unknown. You as an engineer are responsible for the models you create. This includes both their flaws and the models' possible unethical purpose. For example if your employer asks for a model of a missile that is later built and launched, which results in multiple deaths, you would be responsible. Naturally, you are also responsible for any faked data or plagiarism.

A societal impact of simulations becoming more common is the removal of jobs that are no longer needed. The number of workers needed to test a model decreases drastically when virtually all the work is performed by a computer. Furthermore, the kinds of expertise required of the workers changes. On a more positive note, using simulations makes testing safer, which in turn leads to a safer work environment. In the long run, simulations also have the possibility to make societies safer when products can be tested in more ways, and risks can be predicted in a way that would not be possible without simulations.

Simulations of Plasma Creating Electric Wind

Linnéa Sellerholm and Amanda Stenberg

***Abstract*—Plasma actuators are devices that with two electrodes and a dielectric material can ionize the air around it and thus control the airflow. They have considerable potential for a multitude of reasons, one of which being that they have no moving parts, making them easy to produce and hard to break. Using this technology on the front of vehicles like trucks could be revolutionary in increasing fuel efficiency and thus reducing emissions. A model of a plasma actuator in COMSOL Multiphysics was used to simulate the effect it has on the air around it. The focus of the project has been to optimize the design of an actuator for increased velocity in the air around it. This has been done with regards to properties of the applied voltage, the distance between the electrodes and material of the dielectric. Parametric analyses of all the above properties was performed. Close-to-optimal values of some of the above mentioned parameters were successfully calculated. However, other parameters, such as the horizontal distance between the electrodes, were beyond the model's capability to determine using the described method.**

***Sammanfattning*—Plasmaställdon är anordningar som med två elektroder och ett dielektriskt material kan jonisera luften runt sig och på detta sätt styra luftflödet. De har betydande potential av en mängd anledningar, varav en är att de inte har några rörliga delar, vilket gör dem lätta att producera och och svåra att förstöra. Användande av denna teknologi på fronter av fordon som lastbilar skulle kunna vara revolutionerande för ökad bränsleeffektivitet och därmed minska utsläpp. En modell av ett plasmaställdon i COMSOL Multiphysics användes för att simulera effekterna den har på luften runt sig. Projektets fokus har varit på att optimera ett ställdons design för ökad hastighet i luften runt den. Detta har gjorts med avseende på egenskaper hos den tillförda spänningen, avståndet mellan elektroderna och dielektrikumets material. Parametriska analyser för alla dessa egenskaper har genomförts. Nästintill optimala värden för några av de ovan nämnda parametrarna beräknades med framgång. Andra parametrar, som det horisontella avståndet mellan elektroderna, var bortom modellens förmåga att bestämma vid användande av den beskrivna metoden.**

***Index Terms*—Plasma actuators, flow control, dielectric barrier discharge.**

Supervisor: Marley Becerra Garcia

TRITA number: TRITA-EECS-EX-2021:176

I. Introduction

The roads are filled with trucks every day. They are an essential part of both the national and international trading network. Due to their versatility over boats and airplanes, trucks transport everything from food, filling the markets, to other vehicles such as automobiles. European trucks with their large flat front have a high fuel consumption which negatively impacts not only the truck company, but also their customers and above all the environment. Cutting down on drag would directly decrease fuel consumption saving both money and the environment. There are some aerodynamic aspects of the truck where engineers have tried to reduce losses by for example rounding edges but there is still room for improvements.

Plasma actuation is a powerful tool for decreasing aerodynamic losses of trucks and aircrafts. The plasma actuator consists of two electrodes, one embedded within a dielectric material, supplied with a high AC voltage. This device will cause air to ionize thus creating plasma that can impact the air surrounding it. When placed on the A-pillars of trucks this "electric wind" will reduce the turbulence behind the pillars and in turn reduce fuel consumption. This would look like the model in figure 1 where the A-pillars are made out of multiple plasma actuators.

Fig. 1. A truck model for testing plasma actuators at KTH. Source: Taken from [1].

Many geometric designs of plasma actuators have been studied based on the Suzen and Huang model presented in [2]. A more in depth explanation of the different kinds of plasma actuators and how they work can be found at [3]. In this project, simulations have been made based on the Suzen and Huang-model [2]. Different input signals, geometric designs and material constants have been studied, with the goal being to come up with a design suggestion for a plasma actuator.

The project started with getting familiar with the already-developed Suzen and Huang model in COMSOL Multiphysics. This model was first run to perform basic parametric studies of the model parameters. The next step was to investigate how the waveform, frequency and duty cycle of the applied voltage affected the electric field velocity. To analyze the duty cycle an analytical repeating step function was added and used as the waveform for the voltage. Lastly parametric analyses were performed for the horizontal and vertical distances between the electrodes. A short material study was also performed, large with regards to relative permittivity. The latter two studies

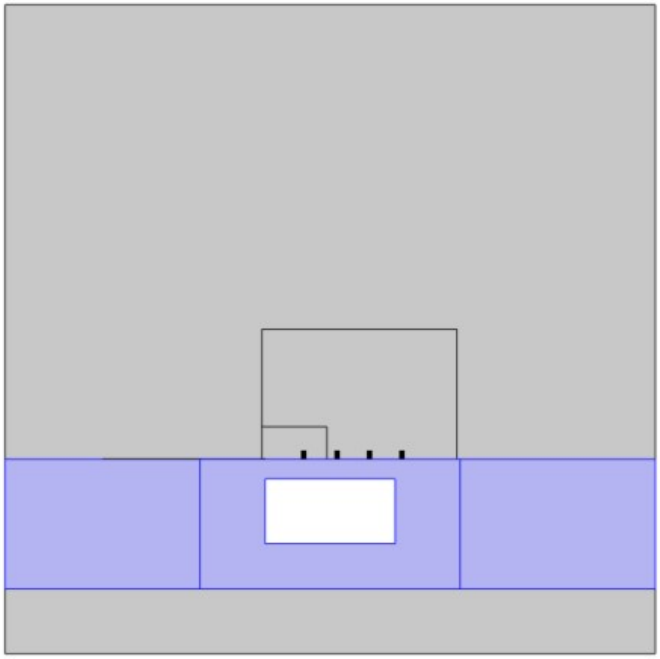


Fig. 2. Model Geometry in COMSOL Multiphysics.

were done with regards to electric energy density due to time constraints. Combining the improved signal, waveform, frequency and duty cycle, and geometric design along with the material choice will make the new model which is to be compared to the original.

II. Method

A. Setup

COMSOL Multiphysics 5.5 was used for all simulations in the project. A simplified, two-dimensional model of a plasma actuator already developed at the Department of Electromagnetic Engineering at KTH was used. The model simulates a multitude of aspects of the plasma actuation, of which the most central to this project was the electric field around the actuator.

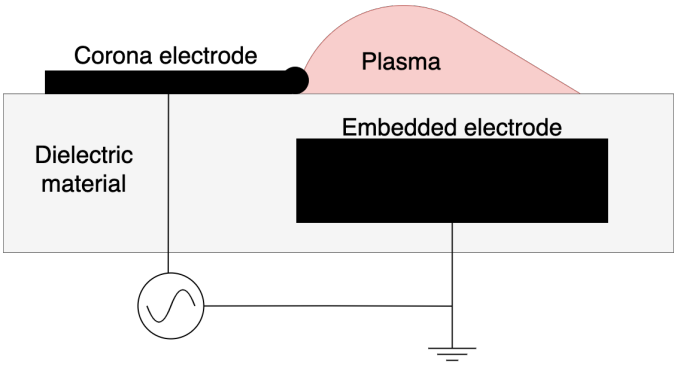


Fig. 3. Diagram of a DBD plasma actuator.

The geometric domains of the model looks as shown in figure 2. The gray domains on the top and bottom of the model are air while the blue domains are the dielectric, an insulating material. The white rectangle in the dielectric represents the embedded electrode. This rectangle domain is unmeshed as the voltage at the boundary of the electrode is the same voltage as the one in the electrode, as it is made of conducting metal. This area is therefore unnecessary to simulate. There is a thin volume on top of the dielectric representing the corona electrode which is connected to the voltage source. This small volume is also unmeshed.

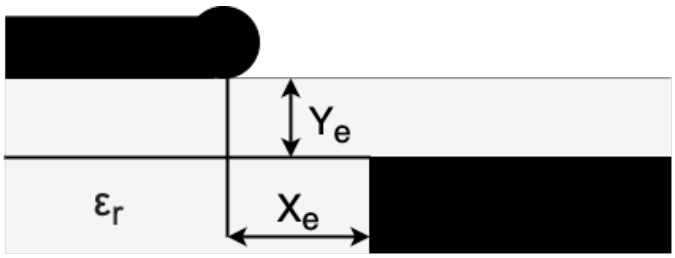


Fig. 4. Horizontal and vertical length between the electrodes illustrated on a diagram of the setup. The figure also includes the relative permittivity of the dielectric.

The model simulates a dielectric barrier discharge (DBD) plasma actuator which creates a low-temperature plasma that can help control the airflow without having any moving parts. A closer look at an actuator shows the corona electrode more clearly, see figure 3. The electrodes receive a high voltage AC signal causing the air to ionize creating the low temperature plasma [2]. The model in COMSOL shows the embedded electrode being fully submerged in the dielectric (figure 2), but in reality the embedded electrode would lie on the surface of the vehicle, connected to ground. All geometric parameters of the electrodes can be adjusted in COMSOL. The ones focused on in the project were the distance between the two electrodes both vertically, Y_e and horizontally, X_e as illustrated in figure 4. The material of the dielectric was studied by changing the relative permittivity, ε_r to correspond to different materials.

TABLE I
Initial parameter values of the model.

Name	Value [Expression]	Unit	Description
V_{max}	9	kV	Peak voltage
λ_d	0.41192	mm	Debye length
ρ_c^{max}	0.0083	C/m^3	Maximal charge density
σ_{Gx}	0.0084076	-	Standard deviation of the Half-Gaussian distribution of the charge density
ω	87965 [$14000 \cdot 2\pi$]	s^{-1}	Angular frequency
α	0.5	-	Duty cycle
X_e	0	mm	Horizontal distance between electrodes
Y_e	3	mm	Vertical distance between electrodes
ε_r	3.1	-	Relative permitivity of dielectric

The given model came with initial parameters as can be seen in Table I, these were working parameters but meant to

be studied and optimized. If nothing else is mentioned these are the parameters used in testing.

B. The Suzen and Huang model

The model which served as the basis for the model used was proposed by Suzen et al. in [2]. This model will henceforth be referred to as the Suzen and Huang model or the SH-model. The SH-model is used to calculate a body force vector (1) that can then be inserted into Navier-Stokes equations to simulate the airflow around the actuator.

$$\vec{f} = \rho_c \vec{E} \tag{1}$$

The body force vector $\vec{f}$ is calculated by simplifying the problem and then solving two equations, one for the electric field caused by the applied voltage (2) and the other for charge density ρ_c (3).

$$\nabla \cdot (\varepsilon_r \nabla \phi) = 0 \tag{2}$$

$$\nabla \cdot (\varepsilon_r \nabla \rho_c) = \rho_c / \lambda_d^2 \tag{3}$$

In order to solve these equations for ρ_c and ϕ some simplifications and assumptions need to be made which will not be explained here. The model uses a Debye length λ_d (the radius of the sphere around a charge carrier in which it's electrostatic effect exists).

The boundary condition for the charge density ρ_c on the dielectric surface, necessary to solve (3), is both time and position dependent as shown in (4). $G(x)$ is an half Gaussian distribution with the standard deviation σ_{Gx} and $f(t)$ is the waveform of the applied AC voltage. In most studies $f(t)$ is a sinusoidal wave [2].

$$\rho_c(x,t) = \rho_c^{max} G(x) f(t) \tag{4}$$

The parameters ϕ and ρ_c are calculated from (2) and (3) using set values of λ_d, σ_{Gx} and ρ_c^{max}. These three values are decided from experimentation to get the model as close to reality as possible. Using the values of ϕ and ρ_c now calculated for each time step, (1) can be solved for every time step as well.

C. Maximizing velocity

The COMSOL Multiphysics model used calculates both stationary and transient simulations. The transient study was most utilized as the main goal was to maximize field velocity. By maximizing the field velocity the ion-neutral momentum will also be maximized, making the actuator more effective [4].

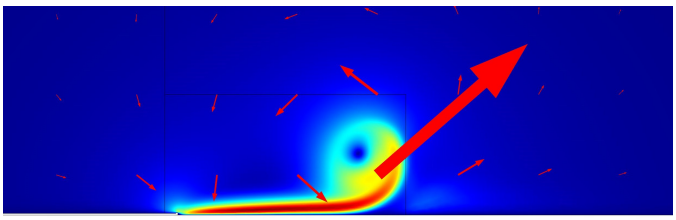
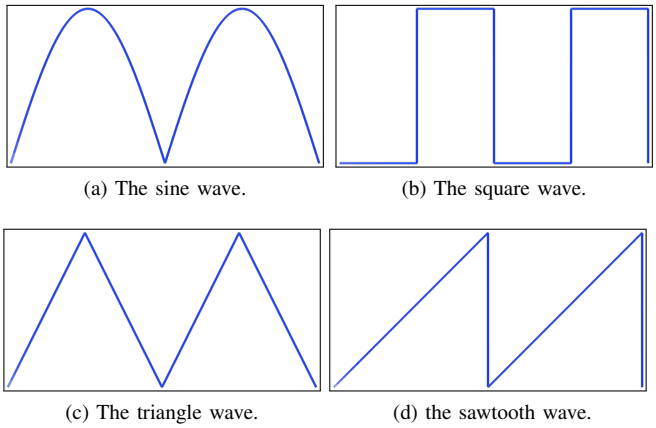


Fig. 5. Surface plot of the velocity field at 5 ms. The surface color represents the magnitude of the velocity, where red is faster. The arrows represents the direction of the field.



(a) The sine wave. (b) The square wave.

(c) The triangle wave. (d) the sawtooth wave.

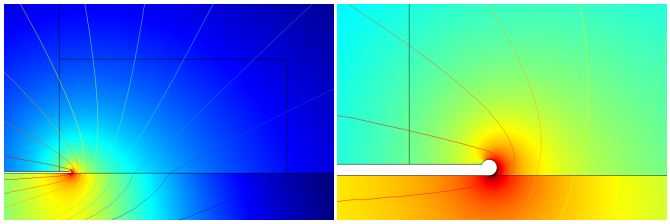
Fig. 6. Four graphs showing the different applied voltage signals after processing with voltage on the y-axis and time on the x-axis. The waves all have the same period and amplitude.

The surface plot as shown in figure 5 represents the velocity field in the fluid. The small white, barely visible stripe on the left side is the corona electrode and the bottom boundary is the dielectric barrier. The embedded electrode is not visible as it is outside the frame. The surface plot was studied to get a more visual representation of the variations in the field, while a regular 2D plot was used to compare values.

The COMSOL function *Waveform* was used to generate different types of AC voltage signals to be applied to the actuator. The waveform was changed between a sine-, square-, triangle- and sawtooth-wave to later be compared in order to find the optimal signal. All signals were modified to go from zero to the maximum voltage V_{max} in order to get a fair comparison to the square wave, which has no negative values. This was done by taking the absolute value of the signal between $-V_{max}$ and V_{max} as well as correcting the period as shown in figure 6. A square wave with a duty cycle of 0.5 was used for the waveform comparison. The effect of the duty cycle was also investigated. The duty cycle analysis was only performed with the square waveform for the values $\alpha = \{0, 0.1, 0.2, \ldots, 1\}$ of the duty cycle. A step function was added to take the voltage value from 0 to 1. The step was then made periodic in a pulse train function that also changed the amplitude to V_{max}. This step function was also used for the sawtooth wave. The angular frequency of the signals were also changed to different values to see how that effected the maximum velocity.

D. Maximizing electric energy density

The choice of geometry and material for the actuator was analyzed to obtain a maximal electric energy density (the in COMSOL built in function es.We), shown in figure 7) integrated over the area closest to the rounded end of the corona electrode. This equals the electrostatic force on this area. All simulations were run with a voltage of $V_0 = 1\,\mathrm{V}$, since the system is linear this made it so that all values can be viewed as being in electrostatic force per voltage with the unit $[N/V]$. The reason the electrostatic force was chosen as the value to optimize for instead of maximum velocity



(a) The area over which es.We was integrated.

(b) Close-up of the rounded edge of the corona electrode.

Fig. 7. Two graphs showing the logarithm of the electric energy density represented by color.

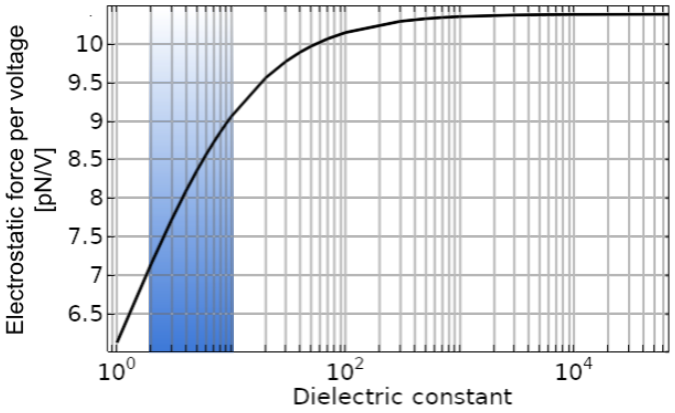


Fig. 8. Variation of the electrostatic force per voltage with the relative permittivity ε_r. All other material properties held constant, done with altered values for geometric parameters.

was due to available computational power, see subsection "Limitations". The calculations for electric energy density and the integration of it had a computational time several orders of magnitude smaller than that for velocity. Running these simulations were thus a useful tool for avoiding unreasonably long computational times.

The impact of the relative permittivity ε_r of the dielectric (also called the material's dielectric constant) was first examined and a parametric analysis was performed. The result is shown in figure 8 where the marked span of dielectric constants are the ones of materials used later in the study. As can be seen in the figure, the energy produced increases drastically for low values of ε_r before the growth slows down to eventually stop almost entirely after $\varepsilon_r = 1000$.

The materials analyzed were chosen for having been studied in this context before (e.g. [4] [5]). All the materials listed in Table II were first tested by their values of ε_r only. This limitation was mainly due to time constraints. From this partial result, materials generating the highest electrostatic force were chosen to use in the velocity simulations.

Geometric parametric analyses were run for values of X_e between $-6\,\mathrm{mm}$ and $8\,\mathrm{mm}$ and values of Y_e between $0\,\mathrm{mm}$ and $6\,\mathrm{mm}$. The middle of these ranges were deemed the most relevant but a few more values were run not to miss any important results. A few tests were run on other geometric parameters of the actuator, such as the size of the electrodes, but since none of these trials gave rise to any larger changes in the simulation they will not be recounted here.

TABLE II
PROPERTIES OF DIELECTRIC MATERIALS [4].

Material	Density ρ $[kg/m^3]$	Relative permittivity ε_r
Teflon™	2160	2.1
Quartz	2200	5
Aluminum Oxide	3700	9.4
Glass	2600	3.8
Lexan™	1190	2.9
Pyrex® Glass	2530	4.1
PC Board	1690	5
Kapton™	1420	3.5
Acrylic plastic	1190	3.1

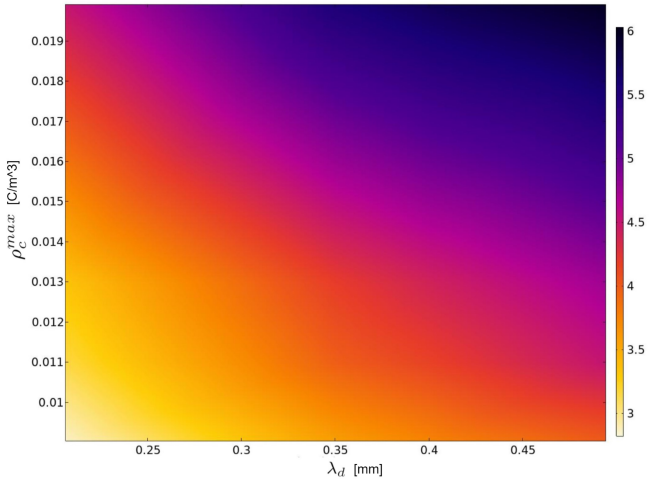


Fig. 9. The maximum velocity [m/s] as a function of λ_d and ρ_c^{max}.

E. Limitations

The major limitation of this project has been the time it took to run the simulations. Due to restrictions and recommendations applying to public transport during the spring of 2021 the project was executed entirely from home. This meant that personal laptops were used for running the simulations. The available computers where a 2019 Razer Blade Advanced, gaming laptop, and a 2015 Apple MacBook Air. The limitation was circumvented by running the more demanding tasks on the Blade and the lighter tasks on the MacBook as the MacBook took a little over three times as long to run a time dependent solver as the Blade. Due to a lack of time, some studies had to be limited to five or seven different data points as some functions were not able to run over night but had to be done manually every other hour.

III. RESULTS

A. Suzen and Huang-parameters

By studying the variables σ_{Gx}, λ_d and ρ_c in the SH-model it was found that ρ_c had the greatest impact on the maximum velocity. It also showed that the greater ρ_c is the greater the velocity. The variable λ_d follows the same pattern, a higher value results in a higher velocity as can be seen in figure 9.

Studying the parameter σ_{Gx} showed that it contributed to very little change. A study of five different values of

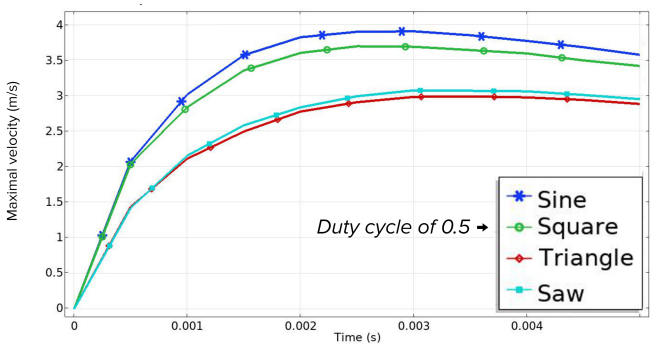


Fig. 10. The maximum velocity of the four waveforms with the same frequency and time step.

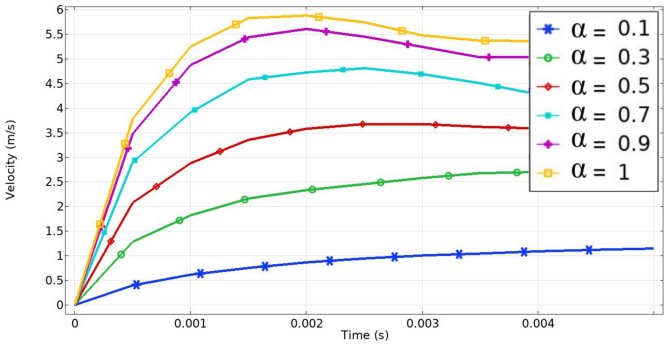


Fig. 11. The maximum velocity of a square wave with different duty cycles but the same frequency and time-step.

σ_{Gx} ranging between 0.009 and 0.2 gave a result where the difference in velocity was around $0.01\,\mathrm{m/s}$, less than 1% of the max velocity.

B. Waveform, frequency and duty cycle

When running the four different wave signals sine, square, triangle and sawtooth, two cases stood out as having lower maximal velocities. These two cases were the triangle- and sawtooth-waves as seen in figure 10. Both the sine- and square-signals had over 20% higher maximum velocities than the other two waveforms.

The sine curve lies a little above the square wave. However a square wave can achieve higher velocities using a different duty cycle as seen in figure 11. The figure shows the results of six different duty cycles where the highest is 1. It shows an increasing velocity if the duty cycle is increased. The maximum value for each duty cycle was taken and and plotted against α. The derivative of this plot was calculated and plotted against the same x-axis, fig 12. This was done to determine the most effective parameter value of α. The top half shows that the relation between duty cycle and velocity is not strictly linear. The second half of the figure shows the derivative of the maximum velocity, $\frac{d}{d\alpha}$. There is a large jump between $\alpha = 0$ to $\alpha = 0.3$, after that the difference is mostly linear until α reaches 0.8 and above.

Testing the two most relevant waveforms with different frequencies shows that the sine and square wave depends on the frequency differently form another. During the frequency

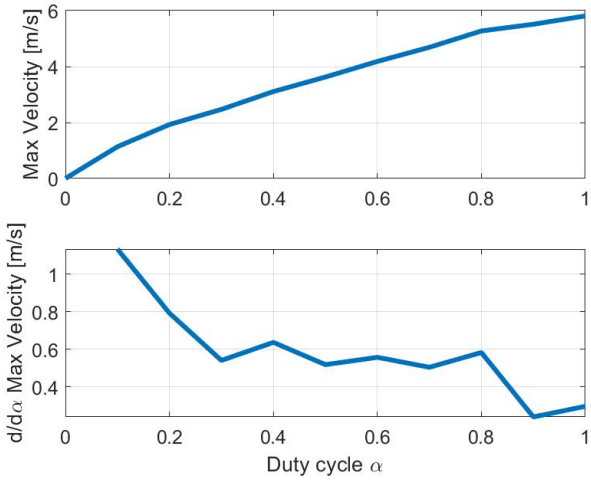


Fig. 12. The top part shows the maximum velocity plotted against the duty cycle value in this case called α while the lower shows the derivative taken from the maximum velocity over α. They are plotted over the same x axis.

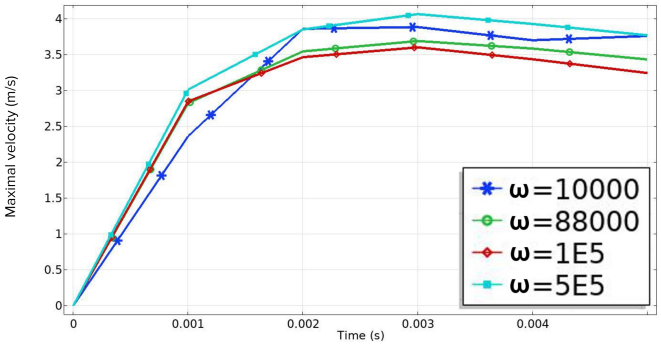


Fig. 13. The maximum velocity for different angular frequencies for a sine wave.

study frequencies ranging from 10kHz to 500 kHz were tested with a sine wave, results were as shown in figure 13. The study was inconclusive as no pattern was found as both the highest frequencies as well as the lowest yielded good results. What was most clear was that lower frequencies gave more unstable curves.

C. Geometry and material

As per the simulations performed, if optimizing for electrostatic Laplacian energy in the immediate area above the dielectric, the distance between the electrodes should be negative, see figure 14. However, using a negative value of X_e did not have the desired effect on the velocity, this will be discussed later in the thesis. The results for the vertical distance Y_e were that it should be as small as possible without being zero. This result seems to, differing from the result for X_e, be accurate with regards to increasing the velocity.

All dielectric materials tested were so based only on their relative permittivity ε_r (figure 15). It was therefore natural that the material that caused the highest electrostatic force per voltage was the material with the highest value of ε_r. That was Aluminium Oxide, Al_2O_3. It being the dielectric material

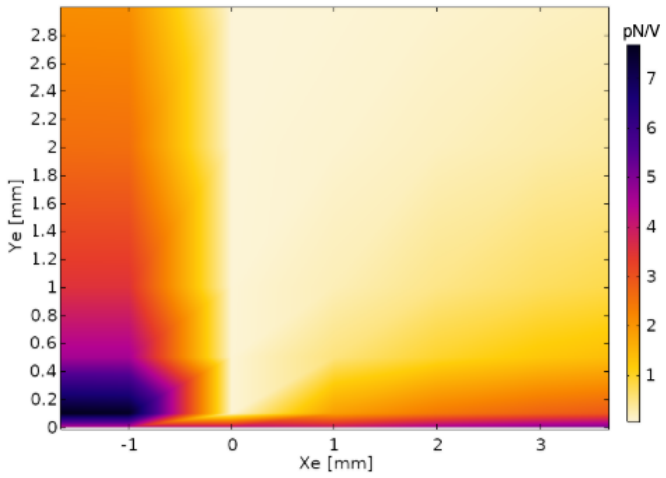


Fig. 14. The electrostatic force per voltage as a function of X_e and Y_e.

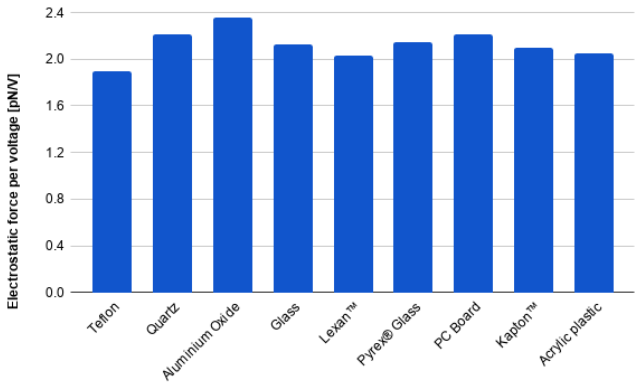


Fig. 15. The electrostatic force for different dielectric materials with original geometric values.

producing the highest electrostatic force it was also tested for velocity with complete material values taken from [6]. The maximum velocity for an actuator with a dielectric of Al_2O_3 was a slight increase from the maximum velocity for the same actuator but with a dielectric of acrylic plastic.

D. Final results

To see if any improvements were made a final comparison was made. A model with the original, given values was compared to a model with improved signal attributes, as well as a model with both improved signal and geometry. The results of the comparison can be seen in figure 16. For the improved models the signal was changed to a square wave with a duty cycle of 0.8 and the vertical distance between electrodes was put at 0.1 mm, this was considered a reasonable small distance.

IV. Discussion

A. Results

Since the purpose of the study study of λ_d, ρ_c^{max} and σ_{Gx} was to get a understanding of the Suzen & Huang model in COMSOL the results were not to be implemented. The parameters are model specific and if they had been changed the model would not represent reality. This makes it so there are no right or wrong results and conclusions.

Comparing figure 10 and 12 the maximum value for a sine wave is around 3.8 m/s, for the square wave to be considered better in that regard a duty cycle above 0.5 would be required. Looking at the derivative in figure 12 the conclusion can be drawn that it is a valid choice as long as the duty cycle does not exceed 0.8 as that would be a waste of energy. Looking at the bottom half of figure 12 there are small bumps at the values 0.6 and 0.8. These would be the better option than 0.7 as you get a higher velocity for a smaller increase in α. These bumps could be eliminated with more data points, but what is certain is that in the midrange the dependence is close to linear.

The frequency study did not yield any clear results as short frequencies gave high velocities for one wave while a high frequency gave a good result for another wave. To get a better result the study would have to be more extensive so that more frequencies could be tested, maybe in a closer range. This could give more clear results. Two possibilities are that the results have a a periodic dependence on the frequency or that a flaw in the model was giving off false positive values for too low or too high frequencies. But when comparing the shapes of the curves, the one for 10 kHz look more unstable than the one for 500 kHz which follows the same smooth shape as the one for the initial value of 88 kHz.

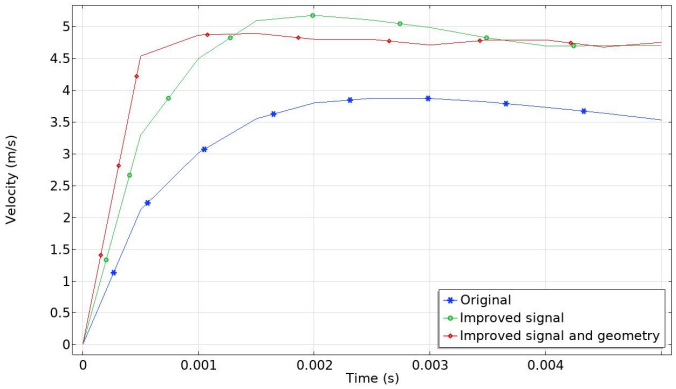


Fig. 16. Maximum velocity of original model compared with improved models.

An idea for how to minimize computational time was to run the geometric and material simulations with regards to electrostatic Laplacian electric energy density. Since that simulation is time-independent unlike the velocity simulation it would take less time and hopefully lead to similar results as with a velocity study. Unfortunately this turned out to not be the case for all parameters. This was predominantly due to the Laplacian field calculations only taking the geometry of the model into account when in reality the voltage supplied to the electrodes and the subsequent space charge have a large impact. Because of this problem the optimized value of X_e for the electric energy density did not align well with previously established theory on the matter of plasma actuator design for increased velocity, see [4]. The optimized value of Y_e does

however not seem to be affected by this problem and instead increases the velocity.

The study has not taken other material properties than the dielectic constant ε_r into account for any other materials than the original acrylic plastic and the aluminum oxide. That fact makes this a very incomplete analysis of the effect the choice of dielectric material has on a plasma actuator. While, with the limited information utilized in the study, aluminum oxide may look like the superior dielectric it is important to consider that important properties of other materials, like their density, would likely change the outcome of this anlysis.

A seen in figure 16 there is a clear improvement in velocity when changing the voltage signal. This supports the conclusions that a square signal with a high duty cycle is effective in increasing velocity. When also adding the change in geometry maximum velocity is reached faster than before and the velocity curve is more stable. At 5 ms the velocity for the improved signal model and the improved signal and geometry model are the same. A higher mean value of the velocity is desirable, and comparing the two by eye their means look quite similar but the one with improved geometry lies just ahead. This also supports the conclusion that a low value for Y_e gives a better performance.

B. Future work

All studies involving the square waveform took longer than other waveforms, this partially due to having to change to a finer mesh to get the program to converge. The finer mesh could have an impact on the results as the finer mesh gives fives times more accurate results at the cost of taking longer. It is somewhat unknown how accurate the model is as it was not possible to do any physical testing, this is something that definitely could be expanded in the future. The model came with simplifications and during the project more simplifications were made to fit the project plan. In the future it would be interesting to test more parameters and see how they effect more than just velocity.

V. Conclusion

Looking at the results it can be concluded that the type of voltage waveform makes a difference, but the deciding factor was the duty cycle which had a strong effect on the velocity. After analyzing the velocities the conclusion was that a square wave with a duty cycle of 0.8 would be optimal. No conclusion about the frequency can be had. The conclusion can also be drawn that a vertical gap distance between the electrodes closing in on zero is optimal. Not much can be said about choices for the horizontal gap distance between electrodes or the material, other than that the relative permittivity ε_r should be high. The final conclusion drawn from this thesis project is that parametric analyses of the time-independent electric energy is not a reliable basis for a study of the time-dependent velocity.

Acknowledgments

We would like to thank our supervisor Marley Becerra for much needed help and endless patience throughout the project. During difficult times in the project Marley was always just a Zoom-call away with necessary information and advice.

We would also like to thank Amanda's cat Elsa, our companion during this project, for bringing us joy during the slower parts of the study especially.

References

[1] D. Callahan. (2017, Mar.) "Plasma could cut wind resistance for trucks". 28.04.2021. [Online]. Available: "https://www.kth.se/en/aktuellt/nyheter/plasma-could-cut-wind-resistance-for-trucks-1.716694"

[2] Y. Suzen, G. Huang, J. Jacob, and D. Ashpis, "Numerical simulations of plasma based flow control applications," in *35th AIAA Fluid Dynamics Conference and Exhibit*. Toronto, ON, Canada: American Institute of Aeronautic and Astronautics, 2012.

[3] K. Adamiak, "Quasi-stationary modeling of the dbd plasma flow control around airfoil," *Physics of Fluids*, vol. 32, no. 8, Aug. 2020.

[4] J. R. Roth and X. Dai, *Optimization of the Aerodynamic Plasma Actuator as an Electrohydrodynamic (EHD) Electrical Device*. Knoxville, TN, USA: AIAA, Jan. 2006. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2006-1203

[5] T. Bonds, D. Sherman, and R. Briggs, "A plasma actuator optimization study exploring the effects of geometric design and dielectric materials selection using thrust and power measurements," in *The 33rd IEEE International Conference on Plasma Science, 2006. ICOPS 2006.* Traverse City, MI, USA: IEEE, 2006.

[6] KYOCERA. (2021, Apr.) "Global Technical Data Alumina (Aluminum Oxide, Al_2O_3)". [Online]. Available: "https://global.kyocera.com/prdct/fc/list/material/alumina/alumina.html"

CONTEXT L – PART I
AIOT: ARTIFICIAL INTELLIGENCE AND THE INTERNET OF THINGS

POPULAR DESCRIPTION

With big data comes big responsibility

Would you rather have privacy or always have a taxi waiting for you when you need one? By collecting large amounts of information, many things become possible. For example, it is possible to predict your transportation needs. However, this comes at a cost. Transferring personal data carelessly conveys a lot of information about you, information which in the wrong hands could be used maliciously.

In only the last 100 years, a lot more data has become available to us than ever before thanks to computers, smartphones and the internet. Great possibilities lie ahead if we can learn how to use this new data.

As we learn to extract more valuable information from data, the demand for large amounts of high quality data will increase. Today this is seen in what is called "The Internet of Things" (IoT), which could be translated as "Connected Things". One example of a popular IoT-device is Amazon's digital assistant, Alexa. These things are connected to each other, and information can be sent back and forth between them. Other interesting examples are smart deodorants and thermostats. Smart deodorants? What do they even do? All we know is that such products actually exist.

While more data has obvious positive effects, there are also great risks associated with it. Over the last few years, we have seen many examples of how personal data has been used unethically. In the Cambridge-Analytica scandal, it was revealed that personal data of many Facebook users was used without their consent. This data was then used in political advertising, which actually made people change their votes in the election. We also see countries like China using facial recognition technology to control their population, restricting freedom of speech and any political opposition.

Another problem with more devices getting connected to each other is the risk of them getting hacked. Say for example, a hacker manages to get through the security of your smart juice press. From there it may be possible for him or her to access other connected devices, such as computers or cellphones. And they are more likely to contain sensitive personal data, such as bank information or medical information. So with buying a juice press comes a small risk of being hacked and getting all your money stolen, so the juice press might turn out to be very expensive.

As the number of IoT-devices and data collection continues to increase, we must remember to put personal privacy and security first. Is it really worth risking your bank information for fresh juice in the morning? The engineers who are developing these products have to make sure they are completely safe from security breaches.

SUMMARY OF PROJECT RESULTS

The increase of Internet of Things-devices has led to an increased demand for Artificial Intelligence, as it is often no longer possible for a human to analyze the big data produced by these devices. Machine learning has made it possible to analyze and extract a lot of information from large, complicated datasets.

The following projects are all applications of machine learning being used on data that has been collected by Internet of Things devices.

Both project groups in L2 focused on human activity recognition (HAR); on the classification of different daily activities of varying intensity (such as sitting, walking, biking and running). Each group implemented two different algorithms to analyse user data collected by smartphone sensors including but not limited to, accelerometers and gyroscopes. The performance of these two algorithms was then compared. Both groups achieved similar results showing that both algorithms could classify the different activities correctly 9 out of 10 times. This could be done with just a few hours of training data. Furthermore, the results of both projects indicated that IoT-data can be useful in raising awareness of how much time we spend each day doing different activities, which in turn can be helpful when making decisions regarding health and wellness.

The project groups in L3 implemented a federated learning algorithm on a pre-existing centralized neural network. The neural network had the purpose of detecting anomalies and minimizing the false alarm rate in water monitoring systems. The data used in the project was collected by a sensor located in the Stockholm archipelago. It measures water temperature, pH value and other values to monitor water contamination. The training of the network was done on a server which collected the data from the sensor, this is the traditional method of training neural networks. Federated learning is a relatively new machine learning paradigm that, unlike the traditional method, trains locally across a collection of decentralized IoT-devices using local data. When the local models have been trained, they are uploaded to a server and combined into a global model, which in turn communicates the new global model to the IoT-devices, this process is then repeated. Both groups implemented their algorithms on virtually simulated sensors, which were constructed by downsampling the data from the only existing sensor discussed above. The result was a program which using a federated learning algorithm and a neural network could predict new measurements from the sensor with 95 percent accuracy.

The activity recognition projects have contributed to the context by producing a service that can take sensory data and translate it into information that is easily read and understood by the user. If the data is collected at a different location than where it will be read by a user, this computation can be done locally and then the result needs only to be transmitted to the device where the user will read it. This reduces the stress on communication networks, as compared to transmitting all data.

The members of the project groups have developed algorithms for processing data, interpreting the data and producing results. A future study could be to implement this into the device that is collecting data, for instance a microcontroller connected to relevant sensors. Another study could be to further develop the program by sending the results from the collecting device to the device that the user will interact with. The result could then be presented to the user in an effective way. Regarding the water monitoring project, further investigation should be put into implementing the developed Federated Learning algorithm using physical sensors.

IMPACT ON SOCIETY AND ENVIRONMENT

With the recent increase in IoT devices, and a projected further increase of these devices, data ethics is an important field to be considered. The rapid expansion of these devices together with the implementation of 5G, speeds up the process of developing well crafted solutions for data handling and transmitting. In this process, it is important to not only consider the economic and technical aspects of new devices, but also the ethical perspective. The possible risks and benefits will be discussed in more detail here from individual, societal and environmental perspectives.

The idea to use sensors to collect data from IoT devices has the potential to increase the life quality at an individual level very much. By analyzing the collected data, it is possible to make more informed decisions in all aspects of life. One example is health: smartwatches and smartphones allow the user to track everything ranging from activity type and distance to pulse and stress levels. As such, these devices can raise the awareness of the user's health, and therefore allow for decisions improving it. Another example relating to the health topic is elderly care. IoT devices such as cameras or other sensitivity equipment can be used to identify falls or other life threatening situations, and would help staff to be able to quickly intervene. Furthermore, IoT devices are helpful in making more efficient decisions as well. For example, traffic jams or other day-to-day occurrences can be detected by smartphones, allowing other road users to make better decisions and save time. However, there are also potential problems with collecting all this data.

One big problem regarding this is the security of the data being collected and transmitted. With many different devices collecting data there are more potential security risks as there are more devices that can be breached. One way to minimize

this risk is by reducing the amount of data being transferred from the devices. This could be done using a federated learning algorithm, as in the water monitoring project, which instead of transmitting all user data, only transmits the locally trained machine-learning models.

The potential environmental impacts of artificial intelligence and IoT are immense. AI- and IoT-technologies can contribute to reduce the overall emissions in the world by applying them on a macroscopic scale or help smaller organizations to cost-efficiently reduce their climate impact. For example, by accurately predicting patterns in sun, wind and waves, AI combined with IoT-devices can improve our systems of managing renewable energy which in turn leads to less carbon pollution.

As the use of IoT devices increases rapidly, a larger amount of data will be produced which in turn will demand more energy since more data needs to be stored. A larger net consumption of energy will have a negative impact on the environment as it can not be promised that the energy will come from renewable sources. However, AI might also be the solution to this problem.

The rapid expansion of IoT-devices does also mean that a lot more electronic products will be developed and produced, which also will have an impact on the environment, as it will use energy and resources, both in production and transportation. It is hard to foresee if the negative impact will be greater than the positive impact that these devices will bring.

One ethical dilemma appears regarding how much information is to be sent between IoT-devices, in association with the risks of being hacked. If a lot of data is sent over these networks, a user might have a better experience with the product, but a hacker might also get his hands on more sensitive information. If less data is sent, the risks are less but the usefulness of the service will probably be worse, so there is a trade-off. Take for instance when IoT-devices are implemented in healthcare and medical devices. Sending more data will likely improve the service and the workflow in a hospital. Are the benefits of better medical devices and increased hospital workflow greater than the potential risk of sensitive data getting into the wrong hands? We think so, as long as cybersecurity is of high importance when developing the systems.

With recent increases in the amount of data being recorded from IoT devices, so is also the concern about anonymity. Even if all the data is collected anonymously from these devices there is still a possible threat of deanonymizing. By analysing different patterns in the data, it is possible to learn more about the data than what was intended. For instance, one might piece together information to find out who the data is coming from, even if the data is sent anonymously. This could lead to infringement on personal integrity, and this risk needs to be assessed when developing services.

Exactly how large of an impact AI will have on the individual, society and environment is hard to say at the moment, but will become apparent in the near future. Overall, we believe that the positive effects of AI and The Internet of Things will outweigh the negatives.

Activity Recogniton Using Accelerometer and Gyroscope Data From Pocket-Worn Smartphones

Oscar Blommegård and Oskar Söderberg

***Abstract*—Human Activity Recognition (HAR) is a widely researched field that has gained importance due to recent advancements in sensor technology and machine learning. In HAR, sensors are used to identify the activity that a person is performing. In this project, the six everyday life activities walking, biking, sitting, standing, ascending stairs and descending stairs are classified using smartphone accelerometer and gyroscope data collected by three subjects in their everyday life. To perform the classification, two different machine learning algorithms, Artificial Neural Network (ANN) and Support Vector Machine (SVM) are implemented and compared. Moreover, we compare the accuracy of the two sensors, both individually and combined. Our results show that the accuracy is higher using only the accelerometer data compared to using only the gyroscope data. For the accelerometer data, the accuracy is greater than 95% for both algorithms and only between 83-93% using gyroscope data. Also, there is a small synergy effect when using both sensors, yielding higher accuracy than for any individual sensor data, and reaching 98.5% using ANN. Furthermore, for all sensor types, the ANN outperforms the SVM algorithm, having a greater accuracy by more than 1.5-9 percentage points.**

***Sammanfattning*—Aktivitetsigenkänning är ett noga studerat forskningsområde som växt i popularitet på senare tid på grund av nya framsteg inom sensorteknologi and maskininlärning. Inom aktivitetsigenkänning använder man sensorer för att identifiera vilken aktivitet en person utför. I det här projektet undersöker vi de sex olika vardagsmotionsaktiviteterna gå, cykla, sitta, stå och gå i trappor (up/ner) med hjälp av data från accelerometer och gyroskop i en smartphone som samlats in av tre olika personer. Två olika maskininlärningsalgoritmer implementeras och jämförs: Artificial Neural Network (ANN) och Support Vector Machine (SVM). Vidare jämför vi noggranheten för de två sensorna, både individuellt och gemensamt. Våra result visar att noggranheten är större när enbart accelerometerdatan används jämfört med att använda enbart gyroskopdatan. För accelerometerdatan erhålls en noggranhet större än 95 % för båda algoritmerna medan den siffran bara är mellan 83-93 % för gyroskopdatan. Dessutom existerar det en synergieffekt vid användande av båda sensorerna, och noggranheten når då 98.5 % vid användande av ANN. Vidare visar våra resultat att ANN har en noggranhet som är 1.5-9 procentenheter bättre än SVM för alla sensorer.**

***Index Terms*—Activity Recognition; Accelerometer; Gyroscope; Data collection; Data Preprocessing; Feature Selection; ANN; SVM**

Supervisors: Afsaneh Mahmoudi Benhangi

TRITA number: TRITA-EECS-EX-2021:177

I. Introduction

In the new era of information, vast amounts of data is being collected and interpreted using a variety of different sensors around us. The most important such device in our everyday life might very well be the smartphone. Modern smartphones have a collection of different sensors including barometer, accelerometer, gyroscope, GPS and proximity sensor, which, among other things, allow for analysis of human behaviour. Human Activity Recognition (HAR) is a widely researched field, focusing on recognizing, or *classifying*, human activity. HAR has a variety of different applications within health, safety and transport optimization [1], [2].

A growing problem in today's society is obesity, which is a major risk factor for several life threatening diseases, including heart disease and stroke [3]. In order to tackle this problem, WHO has issued recommendations regarding physical activity, including that all people older than five years should limit their sedentary time [4]. Hence, a tool for tracking daily activities could potentially be used to raise awareness regarding an individual's sedentary time and help the person to start doing some low to moderate intensity exercise.

In this project, we investigate, and compare, two different machine learning algorithms, Support Vector Machine (SVM) and Artificial Neural Network (ANN), for classifying everyday activities using a smartphone. In total, the following six different everyday activities are examined: walking, biking, standing, sitting, ascending stairs and descending stairs. Also, two different common smartphone sensors, the accelerometer and the gyroscope, are compared in order to determine which of the former, the latter, or both yield the best accuracy for classifying the specified activities.

For the project, no publicly available data set is used. Instead, the data used is collected using a public smartphone application, *CrowdSense*, which allows for recording of both accelerometer and gyroscope data. Furthermore, a process is set up for data preprocessing and feature extraction.

The structure of the paper is as follows. In Section II, data collection and data preprocessing procedures are introduced. Furthermore, the feature selection is discussed, and the machine learning algorithms are described. Also, there is a short description of the performance measures. In Section III, the results are shown. Section IV includes the discussion, and V features the conclusions.

II. Methodology

There are four main parts of classifying human activity recognition using machine learning as specified by [5]. These include data collection, data preprocessing and feature extraction, classification and performance evaluation, as visualized in Fig. 1. These are discussed in greater depths in the following subsections.

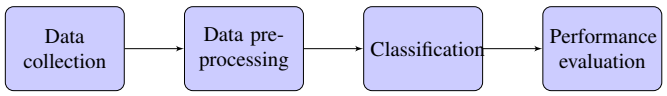


Fig. 1. Common workflow for classification problems.

A. Data Collection

iPhones were used to record the data. Today, most smartphones have a variety of built-in sensors for recording sensory data such as acceleration, altitude and angular momentum. Furthermore, a majority of people has access to smartphones, and tend to carry them with them throughout the day, making the smartphones a good tool for collecting activity data, especially for low to moderately intense activities such as walking, biking, and standing where the subject carries the smartphone. The data in this project was collected with smartphones by using the application *CrowdSense*, a continuous sensing tool available on iOS and Android devices. The data was saved as CSV-files and exported to a database.

Both accelerometer and gyroscope data is frequently used in classifying activities, usually with an accuracy of at least 90 %, as described by [6]–[8]. In our project, the triaxial accelerometer and gyroscope in the iPhones were used to record accelerometer and gyroscope data along all the three axes. Fig. 2 and 3 show examples of how the absolute value of the raw accelerometer and gyroscope data looks like for the six different activities. The data in this project was recorded with a sampling rate of 100 Hz.

In total, three subjects, two male and one female, helped with recording the data. These subjects were instructed to record the data when they were performing the specified activities in their ordinary life. Furthermore, to standardize the data collection process, the placement of the smartphone was in the right or left front pocket of the pants. The reason for this was to reproduce real-life conditions as much as possible. The subjects performed the six different activities as specified in section I: walking, biking, sitting, standing and walking in stairs (up and down). The subjects were further asked to initialize a recording when starting an activity and terminating it immediately after being done, and then label the recording with their name and type of activity.

B. Data Preprocessing

For the raw data from the CSV-files, a certain number of data points in the beginning and end of each measurement was deleted to account for the time it takes for the subjects to put in and take out the smartphones of their pockets. Since the data sampling frequency is 100 Hz, deleting 1 second of data equals 100 data points. The number of seconds deleted depended on the activity, and is shown in Table I.

Data segmentation is the process of dividing the data into smaller subsets, each containing some fixed amount of data points. These subsets are usually referred to as *windows*, In this project, the data was segmented into non-overlapping sliding windows, meaning that no data point was contained in more than one window. Concerning the number of data points in each window, current literature recommends anything between

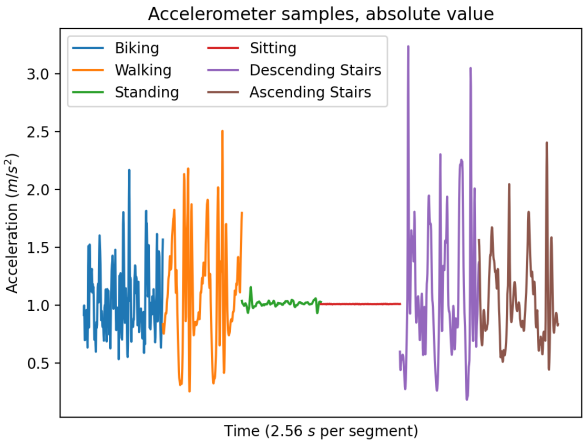


Fig. 2. 2.56 second samples of the absolute value of the accelerometer data for each of the six activities.

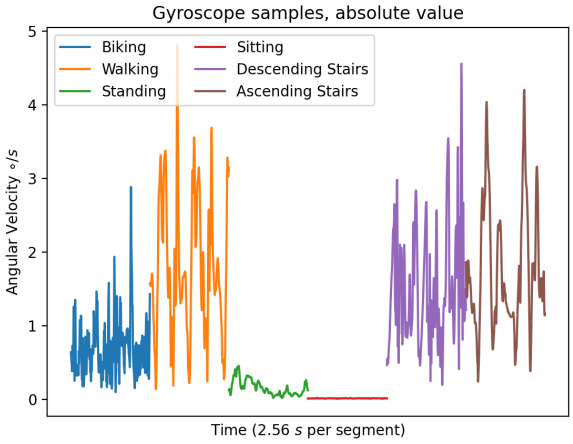


Fig. 3. 2.56 second samples of the absolute value of the accelerometer data for each of the six activities.

$1-15$ s, with longer windows for more complex activities [9]. The data in this project was divided into windows of length 2.56 s, equaling 256 data points.

An important part of data preprocessing is handling imbalances in the data set [10]. This refers to there being very different amounts of data for each type of activity, leading to potential bias in the prediction. Fig. 4 shows the total number of data points for each of the six activities. Walking in stairs (up/down) are clearly the limiting activities regarding the amount of data. The data set can therefore be balanced by limiting the amount of data for each of the other four activities to that of walking in stairs (up/down). Fig. 5 shows the balanced data set. This data set was used in the classification process.

C. Feature Extraction

A critical step when implementing classifiers is extracting features, which are properties of data, from the raw data [6]. In

TABLE I
DELETED DATA POINTS PER ACTIVITY

Activity	Deleted data points (seconds)
Biking	15
Walking	10
Standing	10
Sitting	10
Walking in stairs (up)	5
Walking in stairs (down)	5

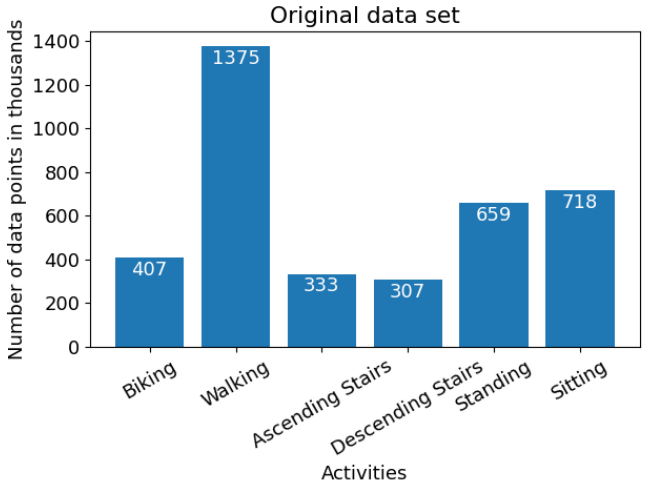


Fig. 4. Total number of data points per activity in thousands for the original data set.

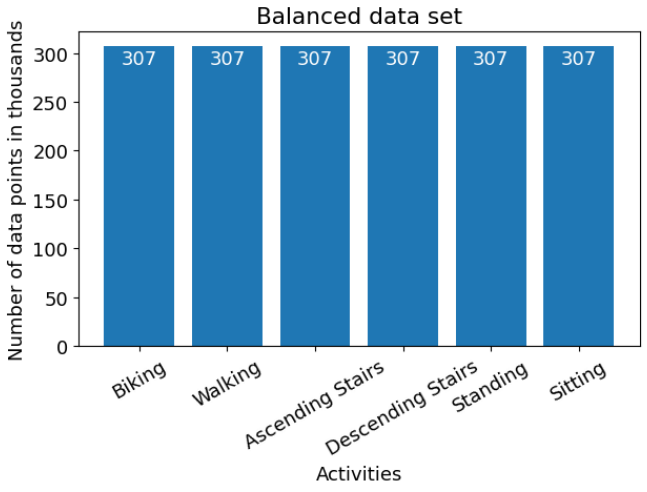


Fig. 5. Total number of data points per activity in thousands for the balanced data set.

this project, the data recordings were divided into 2.56 second windows for which the six features in the list below were extracted. This was done for the acceleration and gyroscope data, which includes the three axes and their absolute value. As a result, a total of 48 time-domain features for each window were extracted. The six attributes are listed below.

- Maximum value: Maximum value for the segment.
- Minimum value: Minimum value for the segment.
- Range: Defined as the difference between the maximum value and minimum value of the segment.
- Mean value: Average value for the segment.
- Absolute mean value: Average of the absolute value for the segment.
- Variance: Variance for the segment.

The attributes above were chosen by selecting attributes that intuitively seemed helpful for recognizing an activity. For example, the average value of the acceleration is probably greater for an activity with greater movement than that of a more still activity, and could therefore serve as a good feature for the machine learning algorithms. Furthermore, many of the features mentioned above are used by [8], [9], [11].

D. Classification

The data was randomly split into two data sets: 80% formed a training set and the rest 20% formed a testing set. Then two different machine learning algorithms were implemented and compared: Artificial Neural Network (ANN), and Support Vector Machine (SVM).

1) **Support Vector Machine (SVM)**: The implementation of the SVM was done using the C-Support Vector Classification algorithm in the publicly available *sklearn.svm* python module. The parameters were chosen as $c = 16$ and $\gamma = 1.05$ through iteration.
2) **Artificial Neural Network (ANN)**: The implementation of the ANN was done using the sequential class in the publicly available *keras.models*. The neural network had four layers including input and output layers, with the two middle layers consisting of 36 nodes each. When training the network, 500 epochs were used.

E. Performance measures

The main performance measurements used is accuracy score. To measure performance, we calculated the total accuracy as

$$\text{Total accuracy} = \frac{\text{Correct classifications}}{\text{Total data points}}. \quad (1)$$

For each combination of data set and machine learning algorithm, the average of ten accuracy measures were taken, each trained on a new random data set split.

III. RESULTS

The total accuracy for the six configurations for the different sensor types is shown in Fig. 6. As can be seen in the figure, using only the accelerometer yielded higher accuracy than using just the gyroscope. The accuracy was roughly twelve percentage points higher using the SVM, reaching an accuracy of 95.6% for the accelerometer compared to 83.3% for the gyroscope. For the ANN, the accuracy for the accelerometer was 98.5% compared to only 89.5% for the gyroscope, which is a nine percentage point difference. Furthermore, using both sensors yielded a slightly higher accuracy for both algorithms.

Moreover, the accuracy using the ANN algorithm is between 1.5-9 percentage points greater than that of the SVM algorithm for all sensors types. Also, Fig. 7-12 presents the confusion matrices for the six activities in the different configurations of data set and machine learning algorithm. Each entry in the matrix represents the fraction of data points of the row's activity that was predicted as the column's activity. Using the accelerometer or using both sensors, the accuracy for all activities except ascending and descending stairs is greater than 97%. For those two activities the accuracy is slightly lower. Using the gyroscope, the accuracies are lower for all activities, but especially for sitting and standing, as well as ascending and descending stairs.

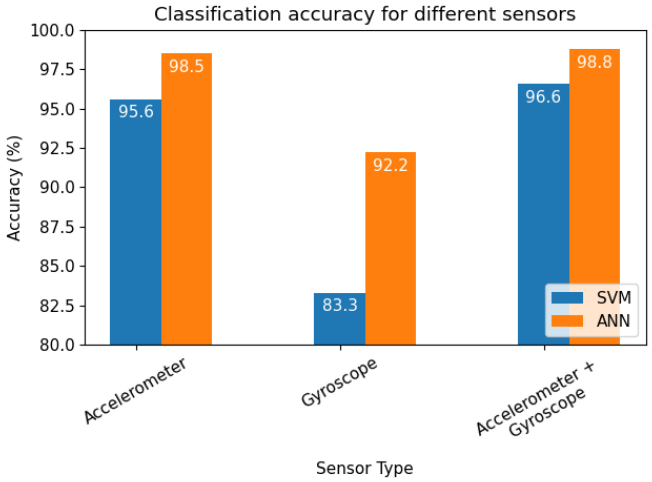


Fig. 6. Accuracy for the different sensor types and for the two machine learning algorithms.

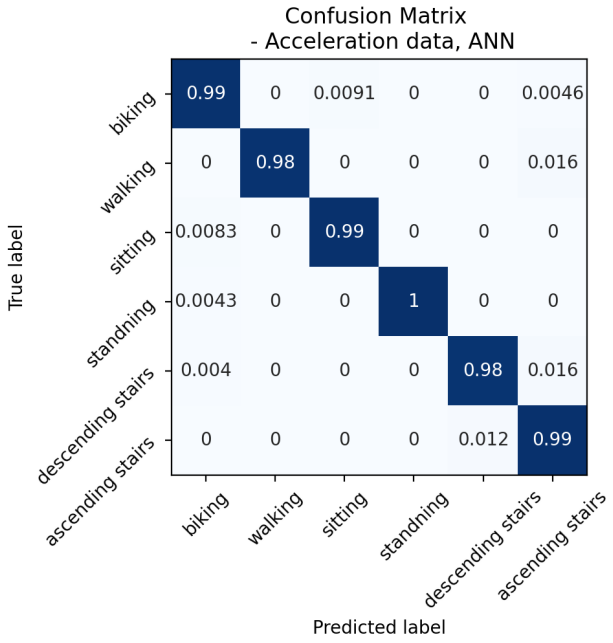


Fig. 7. Confusion Matrix of ANN-implementation on accelerometer data.

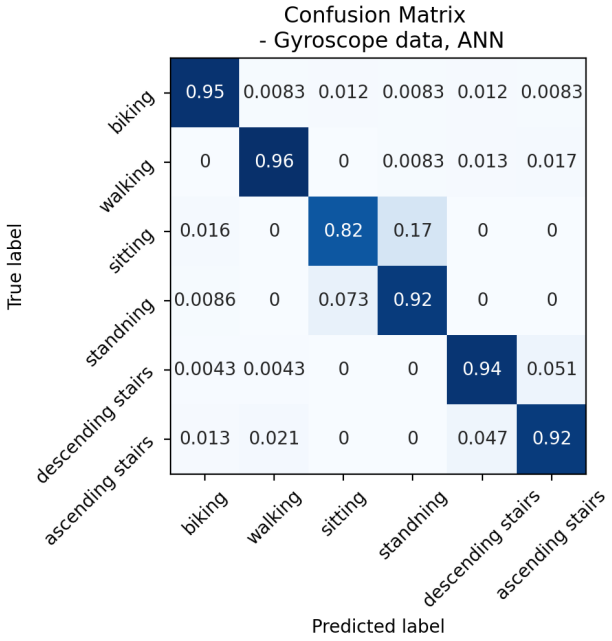


Fig. 8. Confusion Matrix of ANN-implementation on gyroscope data.

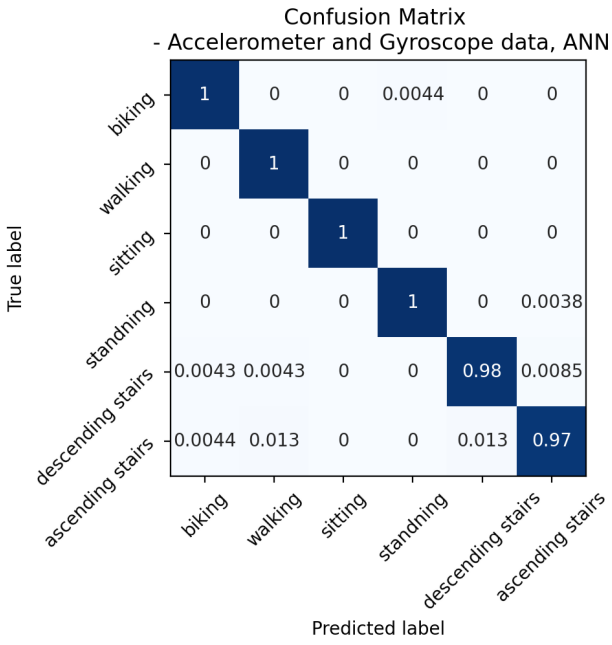


Fig. 9. Confusion Matrix of ANN-implementation on accelerometer and gyroscope data.

IV. Discussion

Our results show that the accuracy is higher using only the accelerometer data compared to using only the gyroscope data. For the accelerometer data, the accuracy is greater than 95% for both algorithms and only between 83-93% using gyroscope data. Also, there is a small synergy effect when using both sensors, yielding higher accuracy than for any individual sensor data, and reaching 98.5% using ANN. Furthermore, for

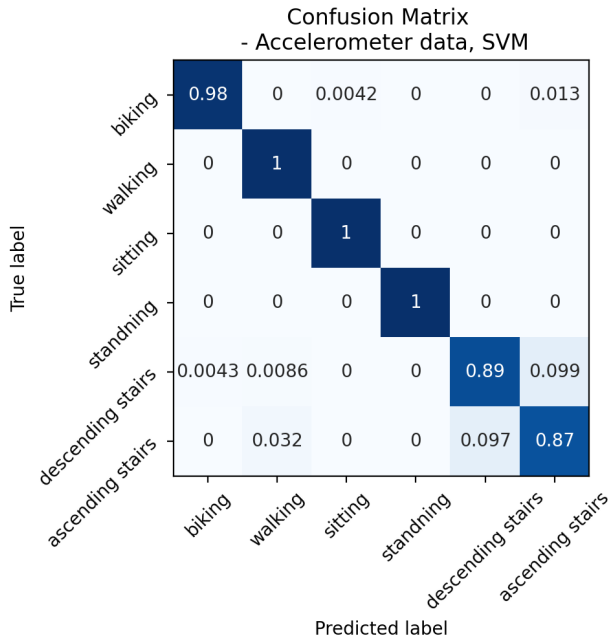


Fig. 10. Confusion Matrix of SVM-implementation on accelerometer data.

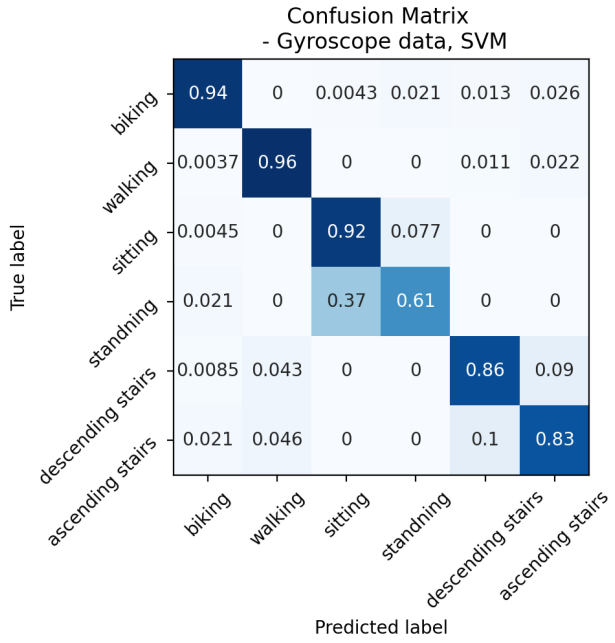


Fig. 11. Confusion Matrix of SVM-implementation on gyroscope data.

all sensor types, the ANN outperforms the SVM algorithm, having a greater accuracy by more than 1.5-9 percentage points. The accuracies are in line with current litterature [6]–[8], even though our results for the accuracy when using the gyroscope data and SVM algorithm seems somewhat lower. We also note that there is a slight tendency to confuse the classification of descending and ascending stairs for both individual senors. This confusion effect is also found in [12]. This could be due to the similar motion patterns of the two

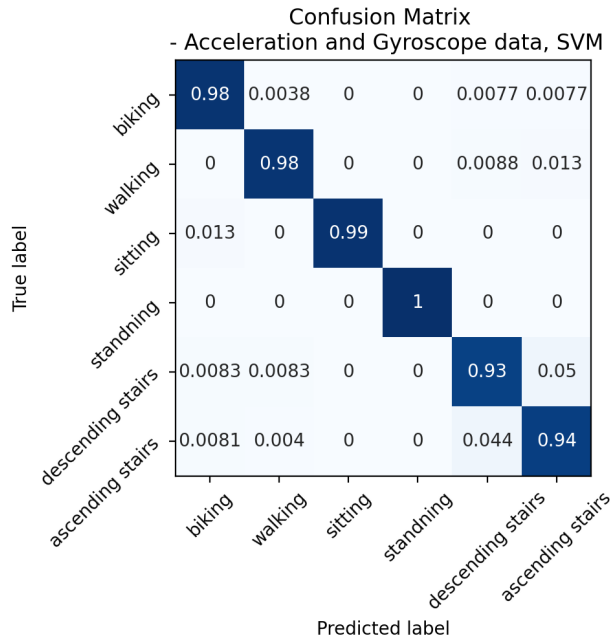


Fig. 12. Confusion Matrix of SVM-implementation on acceleration and gyroscope data.

activities: both are bumpy movements where the smartphone in the subject's pocket rotate with the leg's movement in a similar fashion. A possible way to handle this issue is covered in [13], where analytical transformations are used to extract valuable features. There is also a problem for the classifiers to differentiate between standing and sitting when using the gyroscope data. This is of course due to both activities being very still, resulting in almost no angular velocity.

A. *Limitations*

Since the data was collected when performing activities in the test subjects' ordinary life, there are some smaller disturbances in the data which can make some data segments hard, or impossible, to classify. For example, many stairs have flat sections in the middle. This could give rise to confusion between walking, descending stairs and ascending stairs. Since a few percent of the data might be of this faulty sort, it is questionable whether the inaccuracies of the models are mostly due to the data or the classifiers.

Further, large amounts of the stair data has been recorded in the same staircase. This could make the stair segments easier to classify due to their similarity. Also, all data has been recorded by only three test subjects. Since movement patterns may vary between different persons, a larger test group could result in lower accuracy.

B. *Future research areas*

In the future it would be interesting to examine more features and their relevance in order to optimize the results and the training time of the machine learning algorithms. For example using analytical transformations to extract features as in [13] could improve the accuracy.

V. Conclusion

In conclusion, our research showed that there was palpable difference between using accelerometer and gyroscope data when classifying activities, with the accelerometer being the better choice. There was also a small synergy effect when using both sensors, however, this improvement might have been out-weighted by the fact that two sensors were used instead of one. Furthermore, the ANN outperformed the SVM for all sensor types.

Acknowledgment

The authors would like to thank Afsaneh Mahmoudi Benhangi for her guidance and feedback in this project.

References

[1] G. Mantellos, T. P. Exarhos, and E. Christopoulou, "Human activity and transportation mode recognition using smartphone sensors," in *2020 5th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, 2020, pp. 1–7.

[2] A. Gupta, K. Gupta, K. Gupta, and K. Gupta, "A survey on human activity recognition and classification," in *2020 International Conference on Communication and Signal Processing (ICCSP)*, 2020, pp. 0915–0919.

[3] (2020, Apr.) World health organization. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/obesity-and-overweight

[4] (2020, Nov.) World health organization. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/physical-activity

[5] N. M. Elzein, M. Fakherldin, I. Abaker, and M. AbdulMajid, "Ann-based performance analysis on human activity recognition," in *2019 4th International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE)*, 2019, pp. 1–6.

[6] A. Bayat, M. Pomplun, and D. A. Tran, "A study on human activity recognition using accelerometer data from smartphones," *Procedia computer science*, vol. 34, pp. 450–457, 2014.

[7] N. Tufek, M. Yalcin, M. Altintas, F. Kalaoglu, Y. Li, and S. K. Bahadir, "Human action recognition using deep learning methods on limited sensory data," *IEEE Sensors Journal*, vol. 20, no. 6, pp. 3101–3112, 2020.

[8] Khimraj, P. K. Shukla, A. Vijayvargiya, and R. Kumar, "Human activity recognition using accelerometer and gyroscope data from smartphones," in *2020 International Conference on Emerging Trends in Communication, Control and Computing (ICONC3)*, 2020, pp. 1–6.

[9] G. De Leonardis, S. Rosati, G. Balestra, V. Agostini, E. Panero, L. Gastaldi, and M. Knaflitz, "Human activity recognition by wearable sensors : Comparison of different classifiers for real-time applications," in *2018 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, 2018, pp. 1–6.

[10] P. Shukla and K. Bhowmick, "To improve classification of imbalanced datasets," in *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, 2017, pp. 1–5.

[11] A. Bayat, M. Pomplun, and D. A. Tran, "A study on human activity recognition using accelerometer data from smartphones," *Procedia Computer Science*, vol. 34, pp. 450–457, 2014, the 9th International Conference on Future Networks and Communications (FNC'14)/The 11th International Conference on Mobile Systems and Pervasive Computing (MobiSPC'14)/Affiliated Workshops. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050914008643

[12] R. Yang and B. Wang, "Pacp: A position-independent activity recognition method using smartphone sensors," *Information (Basel)*, vol. 7, no. 4, p. 72, 2016.

[13] M. Ebner, T. Fetzer, M. Bullmann, F. Deinzer, and M. Grzegorzek, "Recognition of typical locomotion activities based on the sensor data of a smartphone in pocket or hand," *Sensors (Basel, Switzerland)*, vol. 20, no. 22, p. 6559, 2020.

Using Machine Learning for Activity Recognition in Running Exercise

Patrik Svensson and Erik Wendel

***Abstract*—Human activity recognition (HAR) is a growing area within machine learning as the possible applications are vast, especially with the growing amount of collectable sensor data as Internet of Things-devices are becoming more accessible. This project aims to contribute to HAR by developing two supervised machine learning algorithms that are able to distinguish between four different human activities. We collected data from the tri-axial accelerometer in two different smartphones while doing these activities, and put together a dataset. The algorithms that were used was a convolutional neural network (CNN) and a support vector machine (SVM), and they were applied to the dataset separately. The results show that it is possible to accurately classify the activities using the algorithms and that a short time window of 3 seconds is enough to classify the activities with an accuracy of over 99% with both algorithms. The SVM outperformed the CNN slightly. We also discuss the result and continuations of this study.**

***Sammanfattning*—Mänsklig aktivitetsigenkänning (HAR) är ett växande område inom maskininlärning då de möjliga applikationerna är stora, speciellt med den växande mängd insamlingsbar sensordata då Internet of Things-enheter blir mer åtkomliga. Detta projekt siktar på att bidra till HAR genom att utveckla två algoritmer som kan urskilja mellan fyra olika mänskliga aktiviteter. Vi samlade in data från den treaxlade accelerometern i två olika smarta telefoner medans dessa aktiviteter utfördes, och satte ihop ett dataset. Algoritmerna som användes var ett faltande neuralt nätverk och en stödvektormaskin, och de applicerades separat på datasetet. Resultaten visar att det är möjligt att med säkerhet klassificera aktiviteterna genom att använda dessa algoritmer och att ett kort tidsfönster med 3 sekunder av data är tillräckligt för att klassificera med en säkerhet på över 99% med båda algoritmerna. Stödvektormaskinen presterade något bättre än det neurala nätverket. Vi diskuterar även resultatet och fortsatta studier.**

***Index Terms*—Activity recognition, HAR, accelerometer, SVM, CNN, mobile sensor**

Supervisors: *Afsaneh Mahmoudi Benhangi*

TRITA number: TRITA-EECS-EX-2021:178

I. Introduction

Machine learning is an area of computer science that dates back to the 1950s [1]. Over the years it has improved considerably and today it can be found in many different areas of application. There are many different types of algorithms used within machine learning. One group of them are supervised learning algorithms which all have in common that their function is to classify novel data by being trained by previous data. Machine learning is a growing field and new applications for machine learning are found every year.

Activity recognition is also a field of growing interest within computing algorithms. It can be used for medical purposes and can for instance be used to recognize if an elderly person is falling to the floor, possibly from having a stroke, and alert medical staff. Collecting data and statistics in the field of sports or exercise enables the user to track progress. It can also be used to track peoples travelling behaviour and used for city planning and public transport planning.

The purpose of this project is to apply machine learning algorithms on sensor data collected from mobile phones to distinguish what the user is doing. Some boundaries had to be set on how many activities the algorithm should be able to distinguish between. The project group decided that the models to be developed should be able to recognize four different activities. The activities the algorithms should be able to distinguish between are running, walking, standing still and walking in stairs.

II. Theory

Two algorithms were chosen for this project for the purpose of comparing the end results. The project group decided to use a support vector machine (SVM) and a convolutional neural network (CNN).

The support vector machine [2], published in 1992, is one of the two algorithms that was used in this project and the other one is a neural network algorithm which concept idea dates back to 1944 according to [3].

A. Convolutional Neural Networks

Neural networks are algorithms whose structure is inspired by the human brain, and [3] describes neural networks as follows. A neural network is made up from many processing nodes which are interconnected in layers. Each layer consists of the processing nodes and different layers may have different amounts of nodes. The data is sent to the first layer, processed and then sent to the next layer and so on until it reaches the final output layer. Data does not travel backwards, rendering this a feed-forward algorithm. The final layer consists of as many nodes as there are possible outcomes.

The processing in each node is done by giving each node input a weight, and calculating the node output from the weighted inputs. All weights in all nodes determine what the algorithm as a whole will output when given an input. The algorithm is trained by first randomizing the weights and then inputting data that has been labelled manually, which is called training data, so the algorithm is able to see if it has arrived at the correct answer. The weights get adjusted during the training process so that the algorithm performs better than it

initially did. The node output is then sent to nodes in the next layer of the network.

In a convolutional neural network, some layers are convolutional layers, where a convolution is applied to the data. Time discrete convolution can be written as

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] \tag{1}$$

where n is the sample, x is the input signal, h is the impulse response and y is the output from the convolutional operation. Convolutional layers are used to extract specific features and recognizing patterns in the data.

A problem when designing neural networks is overfitting. Overfitting occurs when a network is too large in proportion to the complexity of the task to be performed. This results in that the network learns all of the training data's every detail and misses the bigger picture of the problem at hand, often referred to as having poor generalization capabilities. Training the machine model over the training data in high detail results in poor performance over the test data. Test data is the data that gets sent to the algorithm after the training process to validate how well the machine learning model performs over unseen data. There are many different techniques to reduce overfitting. One way is to use a regulator which regulates how the nodes learn. One commonly used regulator is a dropout regulator [4] which is applied on a layer of nodes to randomly suppress the activation of a set percentage of the nodes. This forces the network to learn new pathways to solve the problem which creates a more robust model. However, using the dropout regulators slows down the training process.

B. Support Vector Machine

Support vector machines are the most well known algorithm in a larger class of algorithms known as kernel-based methods, and the concept behind SVMs are, according to [5], described as follows. The idea is to separate values by creating a hyperplane that separates datapoints in the so-called feature input space. The features are features of the data that are relevant for classification. Examples of features for SVMs are mean value and standard deviation, and the features get extracted from the raw data beforehand; the algorithm never sees the raw data. The coordinates for a datapoint in the input feature space are the extracted features that describe the datapoint. The input feature space has a dimensionality of $\mathbb{R}^n$ where n is the number of features, and the hyperplane will have a dimensionality of $\mathbb{R}^{n-1}$ and will divide the feature input space into two subspaces. SVMs are only able to classify between two types of data at a time. It is however possible to make multiple SVMs to solve classification problems with more than two classes.

If the datapoints in the input space are not linearly separable, which is common, the data is mapped by a so-called kernel function to another space where it is linearly separable. This space always has higher dimensionality than the input feature space. It is always possible to find a space where the input data is linearly separable if the dimensionality of the space the data gets mapped into is high enough. It then becomes a problem of finding the hyperplane that is dividing the datapoints in an optimal way by maximizing the distance to the points and handling noisy values as described in [6]. This maximum margin hyperplane is developed through the training process. When the algorithm is exposed to new data after the training process is done, the new datapoints are inputted and the algorithm classsifies every datapoint according to which side of the hyperplane it is placed. The new data has to be handled in the same way as the training data for the algorithm to be effective. If the same features are not identically extracted from the training data and the novel data, SVMs cannot be expected to work well.

As mentioned above, if the data is not linearly separable in the input space, a kernel function has to be utilized which maps the points to another space. A kernel that can be used with SVMs is the radial basis function (RBF) [7] which uses the Gaussian form

$$K(x, y) = e^{-\frac{1}{\sigma^2}||x-y||^2}, \tag{2}$$

where x is the input, y is the target value and σ is the width of the function according to [8]. A relevant parameter for SVMs with RBF kernels is

$$\gamma = \frac{1}{\sigma^2}, \tag{3}$$

as it decides the flexibility of how the hyperplane can be drawn. Too large values of γ cause overfitting which leads to a poor generalization and performance on new data, similar to CNNs. Too small values of γ could cause an ineffective hyperplane fit to the data, resulting in a poor performance accuracy.

Other examples of kernels are linear ones and polynomial ones of different degrees. When choosing which kernel to use for a SVM, the trial and error approach is often the only realistic one according to [9].

To handle noisy or overlapping datapoints, a so-called soft margin [9] are implemented in SVMs to lessen the effect of errors in the data. This effectively means that some datapoints are allowed to be on the wrong side of the hyperplane. It is up to the designer of the SVM to decide how many datapoints are allowed on the wrong side and how far into the wrong side the data are allowed to be. There is a trade-off between the size of the margin and how well SVMs handle errors in the data. The parameter that decides how far on the wrong side the datapoints are allowed to be are decided by the soft-margin parameter according to [10], and it is usually denoted C. The size of the flexibility parameter together with the soft-margin parameter C are of importance when designing a RBF SVM.

Due to the binary nature of SVMs they can only distinguish between two groups of data, in our case two activities. They can however be modified in multiple ways to be able to classify more than two activities according to [5]. One way is to compare data from two activities at a time (one-against-one). Another way is to construct a number of SVMs that is equal to the number of classes to be distinguished, where each compares one class against all other data (one-against-all). This means each SVM answer the binary question if a datapoint belongs to that SVMs class or not according to [9].

A SVM approach will always converge to the same end result when presented with identical train and test data. This is not the case with neural networks, as the weights in the neurons are randomized before each time it trains, meaning that the final trained network might behave differently between executions.

C. Data Processing Theory

Data processing is manipulation of data. A part of data processing is to make sure the data set is balanced. An imbalanced data set occurs when different classes in a data set have different amounts of samples. Training machine learning models on highly imbalanced data sets can lead to the models having difficulties classifying the classes with less samples because of the unequal amount of training samples trained on. One way to balance an imbalanced data set is to use random undersampling. This works by randomly removing samples from the class with the most amount of data until a more balanced data set is acquired. Another technique used for balancing the data is a SMOTE (Synthetic Minority Oversampling Technique) oversampler. SMOTE sampling works by plotting every sample from a class in a space and draws lines in to its closest neighbors. It then creates new synthetic samples placed on the lines, as described in [11].

If magnitudes in the data set varies highly, machine learning models may exhibit problems learning from the low magnitude data values correctly, and could be biased to learning more from the high magnitude data. A scaler or normalization function can be implemented and applied to the input data to prevent this. One way to do this is to subtract the mean value of the input data from the input and dividing it by the standard deviation of the data. This ensures that the standard deviation of the magnitudes are of the same order which will result in better performance from many machine learning algorithms according to [12].

III. Method

Before developing the algorithms, a few steps had to be taken. Initially, a data set had to be obtained which was done by collecting smartphone accelerometer data. Then the data were processed to enable extraction of features that the algorithms would be able to take it as input. The two algorithms were then designed and applied. The programming language used was Python together with the TensorFlow library and scikit-learn as well as support libraries like Pandas, NumPy and Matplotlib.

A. Data Collection

To collect consistent data, a standardized way of collecting the data was developed. The phone was placed vertically in the trouser pocket of the person collecting the data, and the activity were then performed for an unrestricted amount of time and the recordings were stopped afterwards. Two different phones were used by two different people for collecting data. The same data collection app was used with the same collector settings on both devices. The data values collected were the x-, y-, and z-axis from the accelerometer and it was collected at 50 Hz and saved locally in a comma separated value (CSV) file which were later used in the processing. The amount of data collected for each activity can be seen in table I and an example of the data can be seen in fig 1.

TABLE I
Amount of data per activity

Activity	Time collected (minutes)	Number of snippets
Running	65	1300
Walking	56	1120
Standing	48	960
Stairs	53	1060

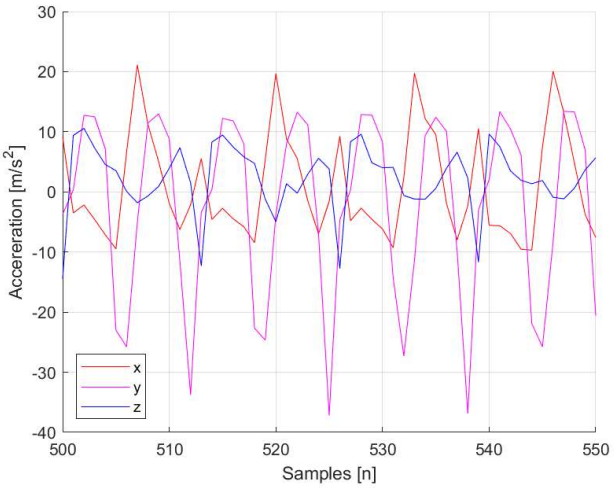


Fig. 1. Example of collected data for one second of the running activity. See appendix for examples of all four activities.

B. Data Preprocessing and Feature Extraction

The data were then preprocessed. The first and last eight seconds from every recorded CSV-file were cut away to account for the time it takes from starting the recording until the phone is in place in the pocket. Then each collected 50Hz datapoint in each file is labeled with the activity it contains and were then merged into one CSV-file.

As mentioned in the theory, the performance will be better with smaller differences in magnitude in the data so a rescaling of the data was made.

As different amounts of data were collected from the different activities, an oversampler using SMOTE was implemented to make up for the unbalanced data. It was set up to create new data samples for the activities with the least recorded data until all activities have the same number of data samples as the one with the most recorded data. This was only applied to the training data so the test data would consist of recorded data without any created data.

All the data in the CSV-file was then portioned into snippets, each snippet containing 200 samples which equals four seconds of data. The initial length of four seconds was chosen arbitrarily. Every snippet that included more than one activity

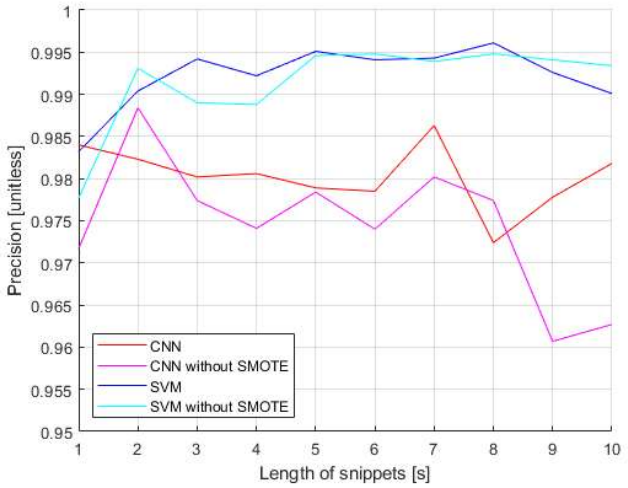


Fig. 2. Graph over the performances with and without SMOTE oversampling using different snippet lengths

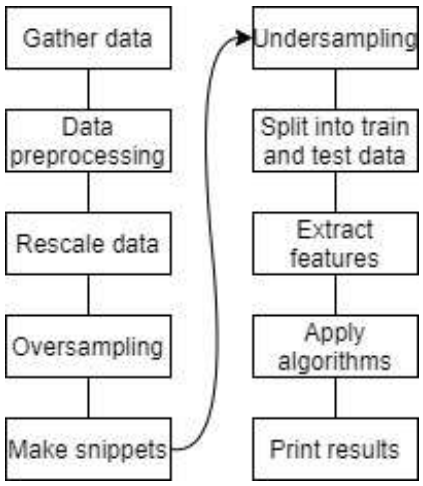


Fig. 3. Block diagram of program

was removed to clean up the data and help the network only associate the right data to its corresponding activity. The best value for the length of the snippets were found later when we had algorithms that were working. It was found experimentally by trying various lengths. The lengths that were tested were 1 to 10 seconds long and the results from the experiment can be seen in figure 2. The graphs that use the SMOTE oversampling were the ones considered when deciding the length, as the oversampling is implemented in the algorithms. As dividing the data into longer snippets gives fewer snippets, a short time was preferred to give more snippets for the algorithms to train and test on, giving higher resolution in the result. Following this logic, the length of snippets for the final algorithms were decided to three seconds long. The length was kept the same for both algorithms to enable an objective comparison of the final result.

An undersampler was implemented to make the amount of snippets equal for each class after some of the values had been removed when combining the samples into snippets. This was only a few samples but it made sure that all the data was completely balanced.

The snippets were then split up into training and test data, with 80% of the data being training data and the rest being test data. Which 20% of the snippets used as test data were decided randomly each time the program were executed. The training and test data were the same for both of the algorithms.

The input to the CNN was the data in the snippets (x-, y- and z-axis acceleration). This is an acceptable input for CNNs as they need strings of information to be able to perform the convolution calculation, see equation (1), in the convolutional layers.

Other features were used for the SVM as the input space would be of very high dimensionality if the raw samples were used and likely to be less optimal due to the algorithm having to map every sample in every snippet via the kernel. Before finding what features would be best for the SVM, a basic SVM was constructed which could take input data and produce a result. Then features were extracted from the snippets and the features were the statistical measures mean value, standard deviation, kurtosis and skew. The values were extracted for each coordinate axis from each snippet. Different combinations of the values were used with the SVM and the results recorded, and it showed that the SVM performed best when only mean value and standard deviation were used as inputs. Therefore, the other features were discarded and mean value and standard deviation were the features used for the final SVM. Because both of the features were extracted for the x-, y- and z-axis separately, the final SVM had six input features in total.

The full flow of data can be seen in figure 3.

C. Algorithm design

The CNN used was a network consisting of two convolutional layers and two artificial neural network layers, which are not convoluting. A dropout was implemented between every layer to reduce overfitting. The output from the first two layers were then transformed to get the right dimension for the rest of the layers. The first, second, third and fourth layers contained 16, 32, 64 and 4 nodes respectively.

For the SVM, the kernel decision was made through trial and error, as recommended in the theory. The kernels that were tried were linear, polynomials of various degrees and RBF and the RBF showed to produce the best results.

To find good values for the flexibility parameter γ and the soft margin parameter C, multiple executions were carried out with different values for the parameters. The test results can be seen in table II. This gave an understanding of what values were appropriate. The same technique were then applied for values around the optimal ones in the table. The highest precision was attained with the parameter values $C = 30$ and $\gamma = 3$.

The algorithms were then considered finished and were executed one last time and the result printed out as confusion matrices.

TABLE II
TABLE OVER PRECISION FOR SOFT MARGIN PARAMETER AND FLEXIBILITY PARAMETER FOR THE SVM

γ \ C	0.1	1	10	100	1000
0.01	90.2%	92.6%	97.1%	97.5%	97.6%
0.1	96.4%	97.3%	97.6%	97.9%	98.4%
1	97.5%	98.1%	98.8%	99.5%	99.2%
10	98.8%	99.4%	99.4%	99.4%	99.4%
100	88.9%	98.3%	98.4%	98.4%	98.4%

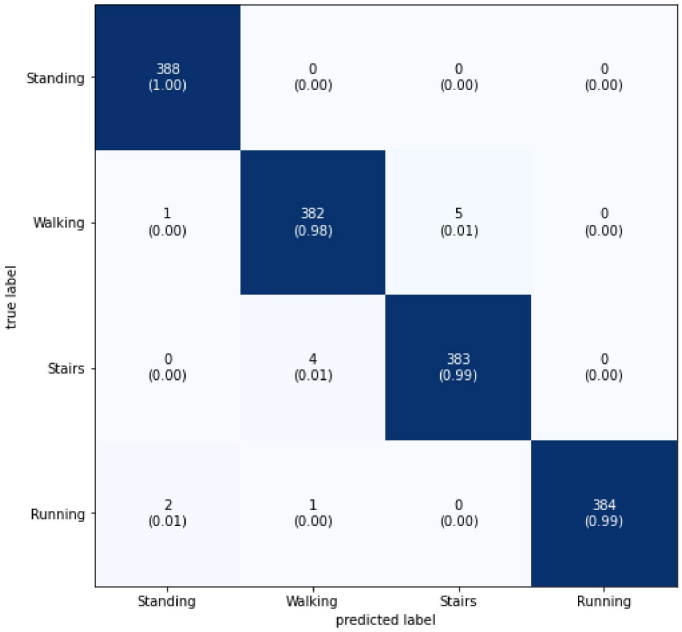


Fig. 4. Confusion matrix for the CNN algorithm

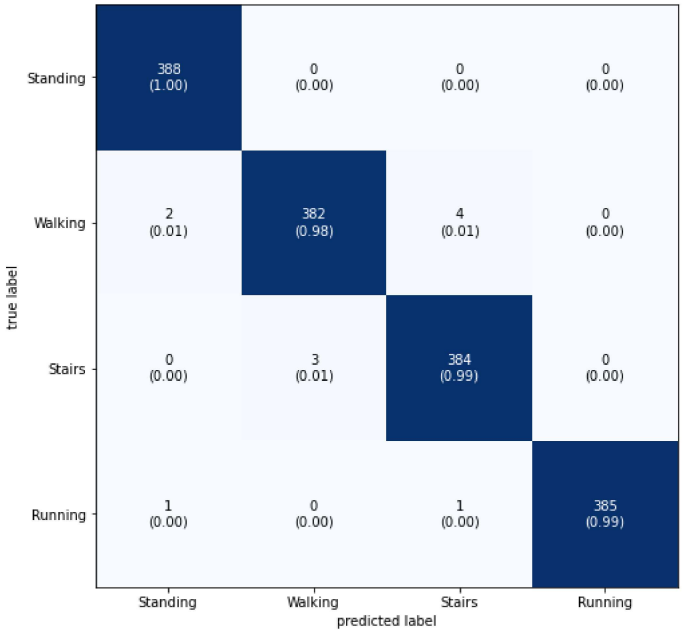


Fig. 5. Confusion matrix for the SVM algorithm

IV. Results

The confusion matrices with the final results can be seen in figure 4 and figure 5. The confusion matrices display how well respective algorithm performed for the four different activities. The rows show what activity a snippet contained was and the columns show what the algorithm predicted it to be, giving a visual way of inspecting how the algorithm performed for each activity. It also shows how many snippets that were wrongly classified and which activities that were most misclassified.

The mean accuracy for the SVM were 99.29% and the mean accuracy for the CNN were 99.16%.

V. Discussion

The results of this project shows that four different activities can be classified with over 99% mean accuracy using machine learning. In this project, the SVM outperformed the CNN slightly. The overall performance was better than anticipated. The high accuracy could be explained by a clean dataset which might be a negative thing for the final products, as the algorithms have not been trained on noisy data and it is not certain how they will perform on imperfect data. Different people have different speeds of walking, running and walking in stairs and collecting data from more than two people is likely to lead to better generalization. It should also be noted that the SMOTE oversampler makes new data points from the existing ones, making the training data even more homogeneous, which might contribute to poor generalization. However, all of the test data were real data with no oversampling applied, implying some generalization in the result. To improve the result, collecting data from another group of people and testing the algorithms on that data would be of interest to see how well the algorithms perform on general data.

Another notable result of this project is that a time window of three seconds was enough to classify the activity with high accuracy, and even one second is enough to classify with relatively high accuracy as seen in figure 2. This implies that it should be possible to design a mobile application that collects the data and continuously computes the activity recognition directly on the smart phone, with a delay of just the length of the snippet plus the short time it takes to portion the data into a snippet and pass it through an already trained algorithm. In the application in running exercise, as was the scope of this project, this real time feature might not be of high importance. But it indicates that it would be possible to develop a mobile application that almost instantly recognizes if the user is falling to the floor, which can be used in elderly care and for alerting medical staff if this were to happen, possibly reducing the number of deaths caused by stroke and other falls in the elderly. Another continuation of this project could be an application that recognizes if a car driver experiences a very sudden retardation, indicating a dangerous impact, and alert medical help.

From figure 2 it is clear that the length of the snippets did not have a significant effect on the accuracy of the algorithms. One thing to note is that because of the random properties of the train test split and the SMOTE sampling the result

will change slightly between executions. The CNN network also has a random nature as mentioned in the theory. This inconsistency in the result could be lessened by executing the algorithms multiple times and calculating the mean of the results for each snippet length.

It can be discussed why the RBF kernel for the SVM performed the best. The explanation could be that it is because of the RBF kernels suitability for classifying non-linear data sets, according to [7]. We also noted that when using different kernels, some values in the confusion matrix barely changed (confusion between running and standing) while others varied heavily (confusion between stairs and standing). This implies that the choice of kernel had lower impact on the values that are relatively separated in the feature input space and higher impact on the values that are overlapping, which is in accordance with the theory.

We found that the algorithms never mistake the activities standing with running or vice versa. This in theory is not surprising, but it shows that the data is likely to not have large amounts of noisy, erroneous values.

There are notable limitations in the result. We know that the algorithms work with collected data from the specific accelerometer data collection app used, and on the two different phones. Problems might arise if one were to generalize this project on other phones or with other apps. Therefore, we cannot draw the conclusion that this accuracy result is relevant for all models of smart phones or with any data collection application.

There are also some methodological limitations as the authors are relatively inexperienced in the area of machine learning and numerical analysis and optimization. A lot of the design choices were made with a trial and error approach. With some parameters, e.g. choice of kernel function, the trial and error approach might be preferred as mentioned in the theory. However, a more methodical approach to the choice of the other parameters might have lead to more optimal values, which would result in better performance. Also, the authors have limited knowledge of the linear algebra mathematics behind machine learning, for instance Hilbert spaces and Lagrangian multipliers, resulting in a less profound understanding of how the algorithms work which might have affected the design decisions and the end result.

VI. Conclusion

This project implemented two machine learning algorithms that accurately classifies human activities from tri-axial accelerometer data when given data from three second long snippets. The methodology was trial and error and iterative optimization to find the parameters for the algorithms. The highly accurate results is possibly explained by a clean and small dataset, which might imply poor generalization. The results show that it is possible to design algorithms that accurately classify between standing, walking, running and walking in stairs with three second long snippets of accelerometer data collected from phones by two people. The SVM outperformed the CNN slightly in this project.

Continuations of this research could be into the medical field or fitness field, by developing applications that can recognize falls or efficiently track exercise data respectively.

VII. Acknowledgement

We would like to thank our supervisor Afsaneh Mahmoudi Benhangi for her help throughout this project.

Appendix A

Examples of collected data for the four activities

References

[1] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development*, vol. 3, pp. 70–105, Oct. 1959.

[2] B. B. et al., "A training algorithm for optimal margin classifier," *In Proc. 5th ACM Workshop on Computational Learning Theory*, vol. 0, pp. 144–152, Jul. 1992.

[3] L. Hardesty. (2017, Apr.) Mit news. Massachusetts Institute of Technology, Cambridge, MA, USA. [Online]. Available: https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414

[4] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for lvcsr using rectified linear units and dropout," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 8609–8613.

[5] C. Campbell and Y. Ying, *Learning with Support Vector Machines*, ser. Synthesis Lectures on Artificial Intelligence and Machine Learning. San Rafael: Morgan Claypool Publishers, 2010, vol. 5, no. 1.

[6] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 273–297, 09 1995.

[7] J. Suriya Prakash, K. Annamalai Vignesh, C. Ashok, and R. Adithyan, "Multi class support vector machines classifier for machine vision application," in *2012 International Conference on Machine Vision and Image Processing (MVIP)*, 2012, pp. 197–199.

[8] M. Mamat and S. A. Samad, "Performance of radial basis function and support vector machine in time series forecasting," in *2010 International Conference on Intelligent and Advanced Systems*, 2010, pp. 1–4.

[9] W. S. Noble, "What is a support vector machine?" *Nature Biotechnology*, vol. 24, no. 12, pp. 1565–1567, Dec 2006. [Online]. Available: https://doi.org/10.1038/

[10] A. Ben-Hur and J. Weston, *A User's Guide to Support Vector Machines*. Totowa, NJ: Humana Press, 2010, pp. 223–239. [Online]. Available: https://doi.org/10.1007/978-1-60327-241-4_13

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[12] F. e. a. Pedregosa, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

Water Anomaly Detection Using Federated Machine Learning

Melker Wallén and Mauricio Böckin

***Abstract*—With the rapid increase of Internet of Things-devices (IoT), demand for new machine learning algorithms and models has risen. The focus of this project is implementing a federated learning (FL) algorithm to detect anomalies in measurements made by a water monitoring IoT-sensor. The FL algorithm trains across a collection of decentralized IoT-devices, each using the local data acquired from the specific sensor. The local machine learning models are then uploaded to a mutual server and aggregated into a global model. The global model is sent back to the sensors and is used as a template when training starts again locally. In this project, we only have had access to one physical sensor. This has forced us to virtually simulate sensors. The simulation was done by splitting the data gathered by the only existing sensor. To deal with the long, sequential data gathered by the sensor, a long short-term memory (LSTM) network was used. This is a special type of artificial neural network (ANN) capable of learning long-term dependencies. After analyzing the obtained results it became clear that FL has the potential to produce good results, provided that more physical sensors are deployed.**

***Sammanfattning*—I samband med den snabba ökningen av Internet of Things-enheter (IoT) har efterfrågan på nya algoritmer och modeller för maskininlärning ökat. Detta projekt fokuserar på att implementera en federated learning (FL) algoritm för att detektera avvikelser i mätdata från en sensor som övervakar vattenkvaliteten. FL algoritmen tränar en samling decentraliserade IoT-enheter, var och en med hjälp av lokal data från sensorn i fråga. De lokala maskininlärningsmodellerna laddas upp till en gemensam server och sammanställs till en global modell. Den globala modellen skickas sedan tillbaka till sensorerna och används som mall när den lokala träningen börjar igen. I det här projektet hade vi endast tillgång till en fysisk sensor. Vi har därför varit tvungna att simulera sensorer. Detta gjordes genom att dela upp datamängden som samlats in från den fysiska sensorn. För att hantera den långa sekventiella data används ett long short-term memory (LSTM) nätverk. Detta är en speciell typ av artificiellt neuronnät (ANN) som är kapabelt att minnas mönster under en längre tid. Efter att ha analyserat resultaten blev det tydligt att FL har potentialen att producera goda resultat, givet att fler fysiska sensorer implementeras.**

***Index Terms*—Federated learning, neural network, anomaly detection, water monitoring, long short-term memory.**

Supervisors: José Mairton Barros Da Silva Júnior, Carlo Fischione

TRITA number: TRITA-EECS-EX-2021:179

I. Introduction

A. Background

Over the last few years, computing power of Internet of Things (IoT)-devices have increased drastically. Previously, the typical IoT-device was just a connection to pass data from one server to another, and all of the heavy computation was done server-side. Today, data processing is being pushed back to the devices. Performing the computations client-side is called *edge computing* [1].

Fig. 1: Visualization of the FL method [3].

With the rise of edge computing, the restrictions of doing computations server-side have become more apparent. For example, one flaw of server-side computing is that it can not be used without encryption, if the collected data is private or sensitive [2]. Locally, this is not an issue.

While the computations generally move closer to the edge devices, new clever algorithms which are able to utilize decentralized data are needed. One of the most influential areas of research regarding computations done locally in IoT-devices is machine learning. Recent research has generated plenty of techniques which tackle the problem in different ways, with one of the most popular being Federated Learning (FL) [4]. It handles the privacy problem by training a machine learning model locally on the collected data, and then only transmitting the fully trained model to a common server for all clients as shown in Figure 1. The local models are then combined into a global model, which is then downloaded to each client as a template for further training. Notice that no raw data is ever sent to the server.

This paper is a part of the iWater project, with partners such as Stockholm City, Ericsson, Stockholm University, SVOA, Telia and Linköping University. iWater is a freshwater monitoring project, with the goal to develop and install a cloud-based water quality monitoring system, using data from connected sensors. The iWater project aims at making water quality data public for all citizens as well as optimizing water quality testing for both drinking and bathing water [5].

Previously, water quality testing was primarily manual, with the analysis carried out in laboratories. This is not nearly as time- and cost-efficient as edge computing with deployed sensors. In EU, water quality is regulated by the Water Framework Directive [6], but it has recently been shown that many lakes and rivers in Sweden do not live up to the requirements presented [7]. The iWater project currently has

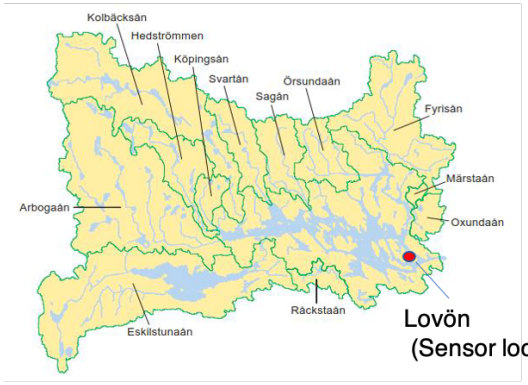


Fig. 2: Location of the water monitoring sensor.

TABLE I: Measurement types and their corresponding labels

Measurement type / unit	Label
Conductivity [μS/cm]	CN
Oxygen saturation [%]	OS
pH-value [1]	PH
Reduction potential [mV]	RX
Salinity [ppt]	SA
Temperature [°C]	TC

one deployed sensor, handled by Ericsson and Stockholm University, as well as a centralized machine learning algorithm that is used to detect anomalies in the water. The development of new Artificial Intelligence (AI) methods and the deployment of new sensors is expected to provide groundbreaking results [8].

B. Problem formulation

FL and IoT-sensors now enables us to create a multi-sensor analysis, with the hopes of creating a network of sensors that are communicating with each other. This could then be used to detect different types of water contamination almost instantly, at multiple locations simultaneously.

In this project, we will investigate whether FL can improve the results produced by the already developed centralized machine learning model in order to provide a basis for whether it would be reasonable, or not, to invest in additional physical sensors.

To fully understand the problem at hand, we will begin at the edge, with the sensor. It is manufactured by Libelium and located in the lake Mälaren. The exact location is shown in Figure 2. It measures the values presented in Table I.

The sensor samples and uploads new data to a server every 20 minutes. The FL method is based on training machine learning models locally on multiple clients. Since we only have had access to one sensor, we have been forced to simulate these clients by splitting the data from the one sensor and distributing it to multiple virtually simulated clients. We will go through this process more in depth in Section III.

The objective is to predict the value of the next measurement made by the sensor based on the earlier measurements using FL and the already developed machine learning model. The idea is that this will make it possible to detect anomalies in the fresh water.

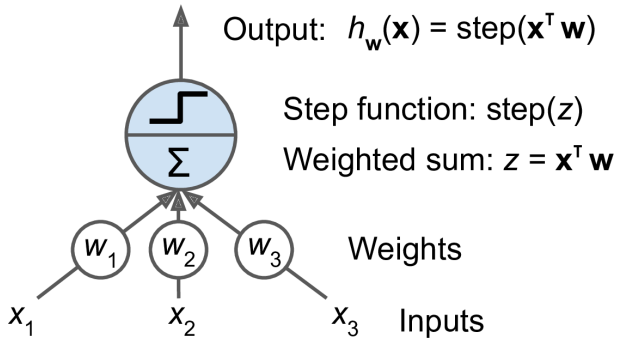


Fig. 3: A perceptron with three inputs and three weights [11].

II. Theory

In this section we briefly present the theory behind Machine Learning and ANNs, and more thoroughly introduce the theory of Long short-term memory (LSTM) networks and FL.

A. Machine Learning and ANN

The definition of a machine learning algorithm is that it is an algorithm able to learn from experience or as formulated by Mitchell [9]:

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E".

In this project the task T is anomaly detection, which translates to finding data that does not fit the rest of the data set, and flagging these as anomalies. The performance measure, P, is the accuracy of these predictions. To measure the accuracy, the data is split into a *training set* and a *test set*. The test set is not used in training, so that we can evaluate how well our model generalizes to data that was not used during training.

ANNs are a class of machine learning algorithms that are inspired by biological neurons. The most basic unit of an ANN is called an *artificial neuron*. An artificial neuron is a function that takes one or more values as input, produces a weighted sum of these values and passes it through a function called an *activation function*, which in turn produces a final output value. Historically, the first artificial neuron was proposed by Rosenblatt [10] and is called a *perceptron*. The inputs and outputs of the perceptron are scalars and each connection is associated with a *weight*, the perceptron described in Figure 3 has three inputs and three weights.

Let $\mathbf{x} \in \mathrm{R}^n$ be the input and $\mathbf{w} \in \mathrm{R}^n$ be the weights. The perceptron computes a weighted sum, z, of its inputs according to

$$z = \mathbf{x}^{\mathbf{T}}\mathbf{w} = \sum_{i=1}^{n} w_i x_i \tag{1}$$

and applies an activation function. The produced output can be expressed as

$$h_{\mathbf{w}}(\mathbf{x}) = step(\mathbf{x}^{\mathbf{T}}\mathbf{w}). \tag{2}$$

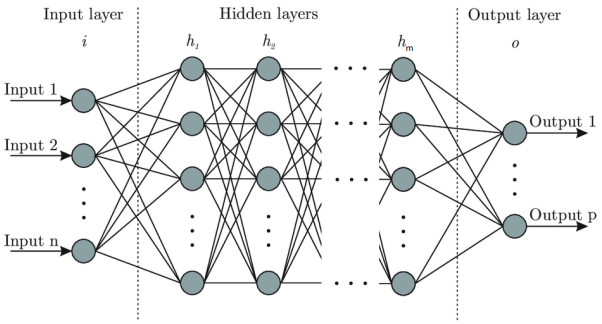


Fig. 4: ANN

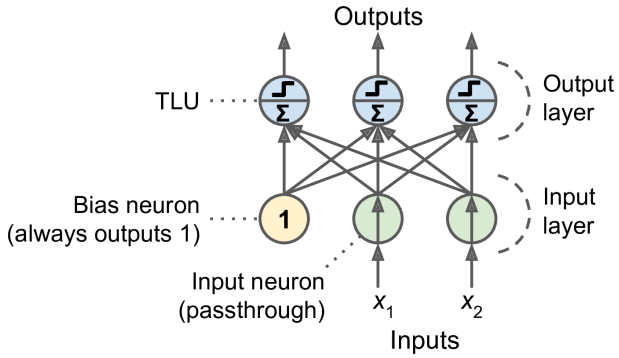


Fig. 5: Layer architecture [11]

The perceptron uses the *Heaviside step function*

$$step(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

as its activation function, which means that it is only able to approximate functions for linearly separable datasets. Modern ANNs use non-linear activation functions in order to be able to approximate functions for more complicated tasks [11]. There are many non-linear activation functions, two of the most frequently used being the Rectified Linear Unit function (ReLu) and the Sigmoid function, see Equation 4 and 5 respectively.

$$ReLu(x) = max(0, x) \tag{4}$$

$$Sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{5}$$

ANNs are built up out of many layers of stacked artificial neurons as shown in Figure 4. It is also common to introduce *bias neurons*, which are neurons that always output a constant. Computing the output of a layer will now look like:

$$h_{\mathbf{W},\mathbf{b}}(\mathbf{X}) = f(\mathbf{XW} + \mathbf{b}). \tag{6}$$

In Equation 6, f(x) is the activation function. **X** represents the matrix of inputs and has one row per input neuron. **W** denotes the matrix of weights, except for the bias neuron. It has one row per input neuron and one column per neuron in the layer. **b** is the bias vector which contains all connections between the bias neuron and the regular neurons, it has one bias term per neuron. There is always one bias term per neuron as shown in Figure 5 [11]. The name *bias* originates from the fact that the output is biased towards having the value b if there does not exist an input [12].

The objective of a neural network is to approximate a function ϕ. In short, a *feed forward neural network* defines a mapping $y = \phi(\mathbf{x}; \theta, b)$ and learns the values for the weights θ and biases b that minimizes the error of the approximation. The error is measured using a *loss function*, which is a function that computes the error between the output of the neural network and the given target value. Feed forward neural network are networks where the connections between the neurons do not form cycles, we will discuss other types of neural networks in the following subsection.

One example of a loss function, and the one that is used in the centralized machine learning model, is the cross-entropy loss function. It is especially useful when classifying labeled data. The output is given as a probability between 0 and 1. As the actual predicted probability diverges from the intended label, the loss function increases. In other words, it penalizes predictions which are both wrong and confident the hardest. More rigorously, the cross-entropy loss function L is defined as:

$$L = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}). \tag{7}$$

Data binning is a commonly used data-quantization technique where the original data values that fall into a given interval, a *bin*, are replaced by a value representative of that interval. We will elaborate on how this technique works and how it connects to our project in an upcoming section. In the equation above, M is the number of bins, y is an indicator, which is either 0 or 1 depending on if the bin label c is the right classification for the observed data o and p is the predicted probability.

The learning is done using the *Backpropagation algorithm* and *Stochastic gradient descent* (SGD) [12]. The gradient of the loss function is calculated using the Backpropagation algorithm. It does this by computing the partial derivatives of the loss function with respect to each weight and bias. SGD is a commonly used optimizer that uses the gradient of the loss function in order to determine how to update the weights and biases such that the error between the output and the target value is minimized. The so-called learning rate parameter η controls how much the weights and biases of the ANN changes in response to the loss function after an update. Choosing a value that is too large may result in instability or learning a set of sub-optimal weights and biases too quickly, while setting it too low might result in the training taking a very long time.

SGD is not the only commonly used optimization method, the *Adam optimizer* is another example of a frequently used optimizer. In short, it uses momentum and an adaptive learning rate to converge faster, it is explained in more detail by Kingma and Ba in the article *Adam: A method for stochastic optimization* [13].

To reduce the risk of over-fitting a neural network, L1-regularization is a commonly used method. It works as an

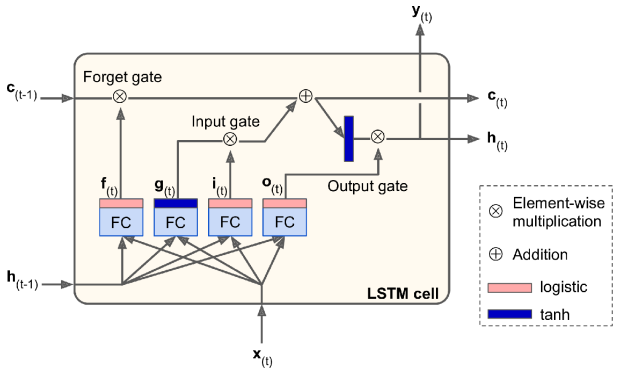


Fig. 6: Architecture of a LSTM cell [11].

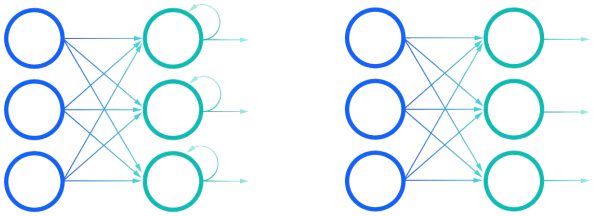


Fig. 7: Illustration of a RNN (left) and a feed forward neural network (right), respectively [17].

extra term in the loss function and reduces some weights to zero, producing sparse models [11].

We will now introduce the terms *batch-size* and *epoch*, both of which will be used frequently throughout this report. Data is typically passed through an ANN in so called batches, or smaller subsets of the entire dataset. The batch-size is the total number of training examples present in one batch. One epoch is done when an entire dataset is passed through an ANN exactly one time.

B. Long short-term memory

The most common and effective way to train long, sequential data is using a *recurrent neural network* (RNN) with LSTM, the difference between a feed forward neural network and a RNN is illustrated in Figure 7. LSTM has become the new state of the art for solving many previously difficult problems. This includes but is not limited to translation [14], speech modeling [15] and audio analysis [16].

The reason for LSTM-networks being effective is that the cells, which are the building blocks that make up the layers in an ANN, can keep track of their state over time, as well as having a *forget gate*. To show what this means we will follow the explanation formulated by Géron [11]. The cell is made up by three gates: an input gate, an output gate and a forget gate as, shown in Figure 6. Observing the figure whilst reading the explanation below is highly recommended. The long-term state $\mathbf{c}_{(\mathbf{t-1})}$ first goes through the forget gate, which makes it drop some memories. In the next stage it acquires new memories, chosen by the input gate. The resulting state vector $\mathbf{c}_{(\mathbf{t})}$ is then sent out of the network. However, after the addition of memories the long-term state vector is also copied, passed through the *tanh* function and filtered through the output gate. This whole procedure creates the short-term state $\mathbf{h}_{(\mathbf{t})}$, which is equal to the output $\mathbf{y}_{(\mathbf{t})}$. The input vector at time *t*, $\mathbf{x}_{(\mathbf{t})}$, and $\mathbf{h}_{(\mathbf{t-1})}$ are then fed to 4 different layers, the main layer outputs $\mathbf{g}_{(\mathbf{t})}$. The most important parts of this output is stored in the long-term state vector, and the rest is dropped.

The three other layers are all *gate controllers*. Their outputs are ranging from zero to one. A zero represents a closed gate and a one represents a gate being open. The *forget gate* decides which parts of the long-term state should be deleted, the *input gate* decides which parts of $\mathbf{g}_{(\mathbf{t})}$ should be added to the long-term state and the *output gate* decides which part of the long-term state that should be output in the present time step. We define the weights for a LSTM-layer as:

Input Weights: $\mathbf{W_{xi}}, \mathbf{W_{xf}}, \mathbf{W_{xo}}, \mathbf{W_{xg}}$.

Recurrent Weights: $\mathbf{W_{hi}}, \mathbf{W_{hf}}, \mathbf{W_{ho}}, \mathbf{W_{hg}}$.

Bias terms: $\mathbf{b_i}, \mathbf{b_f}, \mathbf{b_o}, \mathbf{b_g}$.

This gives us the following equation for how to compute the long-term state, the short-term state and the output at every time step:

$$
\begin{aligned}
\mathbf{i}_{(\mathbf{t})} &= \sigma(\mathbf{W}_{\mathbf{xi}}^{\mathbf{T}}\mathbf{x}_{(\mathbf{t})} + \mathbf{W}_{\mathbf{hi}}^{\mathbf{T}}\mathbf{h}_{(\mathbf{t-1})} + \mathbf{b_i}),\\
\mathbf{f}_{(\mathbf{t})} &= \sigma(\mathbf{W}_{\mathbf{xf}}^{\mathbf{T}}\mathbf{x}_{(\mathbf{t})} + \mathbf{W}_{\mathbf{hf}}^{\mathbf{T}}\mathbf{h}_{(\mathbf{t-1})} + \mathbf{b_f}),\\
\mathbf{o}_{(\mathbf{t})} &= \sigma(\mathbf{W}_{\mathbf{xo}}^{\mathbf{T}}\mathbf{x}_{(\mathbf{t})} + \mathbf{W}_{\mathbf{ho}}^{\mathbf{T}}\mathbf{h}_{(\mathbf{t-1})} + \mathbf{b_o}),\\
\mathbf{g}_{(\mathbf{t})} &= \tanh(\mathbf{W}_{\mathbf{xg}}^{\mathbf{T}}\mathbf{x}_{(\mathbf{t})} + \mathbf{W}_{\mathbf{hg}}^{\mathbf{T}}\mathbf{h}_{(\mathbf{t-1})} + \mathbf{b_g}),\\
\mathbf{c}_{(\mathbf{t})} &= \mathbf{f}_{(\mathbf{t})} \otimes \mathbf{c}_{(\mathbf{t-1})} + \mathbf{i}_{(\mathbf{t})} \otimes \mathbf{g}_{(\mathbf{t})},\\
\mathbf{y}_{(\mathbf{t})} &= \mathbf{o}_{(\mathbf{t})} \otimes \tanh(\mathbf{c}_{(\mathbf{t})}),
\end{aligned}
\tag{8}
$$

where $\sigma(x)$ is the logistic function and $\otimes$ denotes element-wise multiplication.

This is the most commonly used version of a LSTM cell, but there are many other variants. The most common modification of the vanilla LSTM cell is the *gated recurrent unit* (GRU), which has a coupled input and forget gate.

C. Federated learning

There is a vast amount of data which is generated on IoT-devices around the world every day. Traditionally when training ANNs, or mathematical models in general, this data is collected by a centralized server, which also executes training.

FL proposes an approach where a set of clients learns locally and only sends an updated model to a global server. The global server learns a shared global model by aggregating the locally computed weights and biases. We will present a FL algorithm called FederatedAveraging (FedAvg) which was developed by Google researchers [4] and has shown promising results.

When developing any machine learning algorithm, we usually make the assumption that examples in each dataset are *independent* and *identically distributed*. If these assumptions

are met we call the data *IID*, but in a FL setting this is rarely the case. However, it is shown by McMahan et al. [4] that good results can be achieved even if the training data is non-IID.

Assume that there exists a global server and that there are K clients, each with a fixed local dataset. Consider the following neural network loss function

$$\min_{\omega \in R^d} f(\omega) \text{ where } f(\omega) := \frac{1}{N} \sum_{i=1}^{N} f_i(x_i, y_i, \omega), \tag{9}$$

where $f_i(\omega)$ is the loss of the prediction on example (x_i, y_i) given model parameters ω.

Let $\mathcal{P}_k$ refer to the dataset stored on client k and let n_k=$|\mathcal{P}_k|$. Equation 9 can now be written as

$$\begin{aligned} f(\omega) &= \textstyle\sum_{k=1}^{K} \frac{n_k}{n} F_k(\omega), \\ F_k(\omega) &= \frac{1}{n_k} \textstyle\sum_{i \in \mathcal{P}_k} f_i(x_i, y_i, \omega). \end{aligned} \tag{10}$$

Let g_k=$\nabla F_k(\omega_t)$ and let η be the learning rate. FedAvg initiates with the random selection of a fraction C of the clients, then the server communicates the current global state to each of the clients.

The selected clients proceed to perform their local training based on the global state and their local datasets. Their updates are then sent to the server, which in turn aggregates all the local updates and applies these to its global state.
The local model parameters are updated according to

$$\omega_{t+1}^k \leftarrow \omega_t^k - \eta g_k \; \forall k, \tag{11}$$

and then aggregated to the central server according to

$$\omega_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} \omega_{t+1}^k. \tag{12}$$

The process is then repeated.

In other words, for each global step, each client trains locally using its local data and the server then takes a weighted average of the resulting local models. Note that when aggregating the local updates, the new updated weights are weighted to be proportional to the number of data samples for client k as n_k/n. This makes sure that clients with more data have a larger impact on the aggregated global model. The entire algorithm is described, in pseudo-code, in Figure 8.

Algorithm 1 `FederatedAveraging`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:
initialize w_0
for each round $t = 1, 2, \dots$ **do**
  $m \leftarrow \max(C \cdot K, 1)$
  $S_t \leftarrow$ (random set of m clients)
  for each client $k \in S_t$ **in parallel do**
    $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
  $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$

ClientUpdate(k, w): *// Run on client k*
$\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)
for each local epoch i from 1 to E **do**
  for batch $b \in \mathcal{B}$ **do**
    $w \leftarrow w - \eta \nabla \ell(w; b)$
return w to server

Fig. 8: Pseudo-code describing FedAvg [4].

Fig. 9: Illustration of centralized machine learning model with one LSTM-layer.

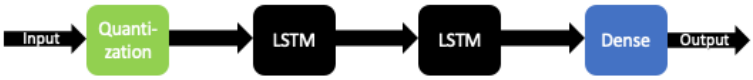


Fig. 10: Illustration of centralized machine learning model with two LSTM-layers.

III. Method

The centralized machine learning model will be described in Section III-A, the various software frameworks used in the project will be described in Section III-B, the process of simulating sensors will be explained in section III-C, the implementation of the FedAvg algorithm will be presented in section III-D and our choices regarding producing results will be presented in III-E.

A. Centralized machine learning model

The input, I of the centralized machine learning model is:

$$\begin{aligned} I &= TC + X, \\ X &\in \{SA, CN, RX, PH, OS\}. \end{aligned} \tag{13}$$

The temperature measurement will always be part of the input, the reason being that temperature highly correlates with all the other measurement types, hence we want our model to always take the temperature into account.

The output is the prediction of in what range the next measurement of X will be in. This means that we have to split the prediction range into bins. We are using a so-called quantile cut function in order to put all historical data into 6 equally large bins. We used 6 bins since this gave us a good balance between the ranges associated with the bins and the accuracy of the predictions.

Figure 9 and Figure 10 illustrate our model with one and two LSTM-layers, respectively.

B. Software frameworks

The centralized machine learning model had been implemented in *Python* using *Tensorflow 1.13*, which is a machine learning library developed by Google. It was therefore natural for us to also use these technologies in this project, all the code developed in this project is available on GitHub [18]. All data from the sensor has been available for us to use through *MongoDB*, which has allowed us to easily download the most up-to-date data for training.

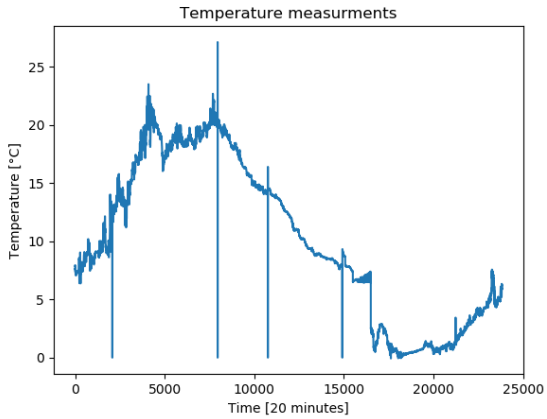


Fig. 11: Raw temperature data collected by the sensor during a time period of 365 days.

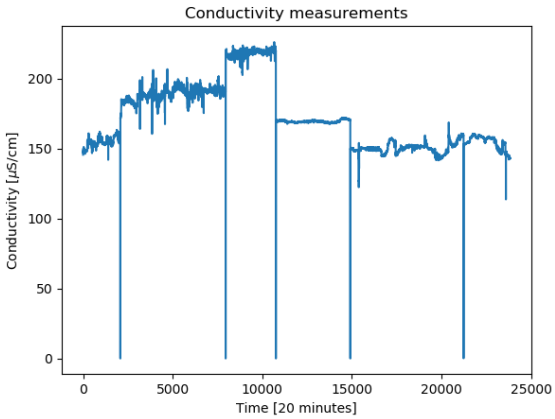


Fig. 14: Raw conductivity data collected by the sensor during a time period of 365 days.

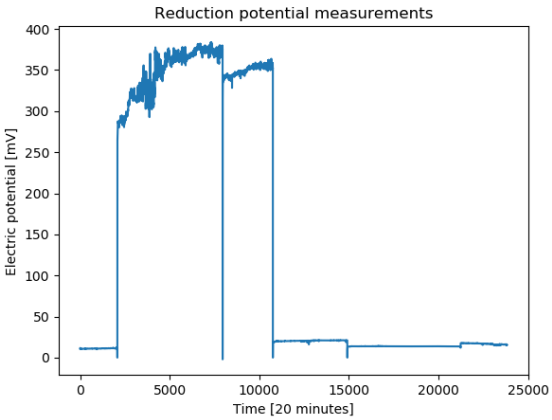


Fig. 12: Raw reduction potential data collected by the sensor during a time period of 365 days.

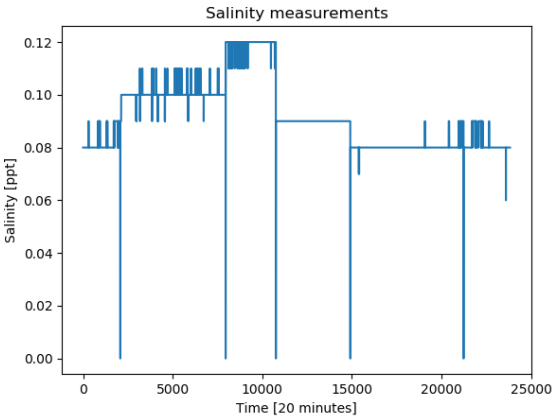


Fig. 15: Raw salinity data collected by the sensor during a time period of 365 days.

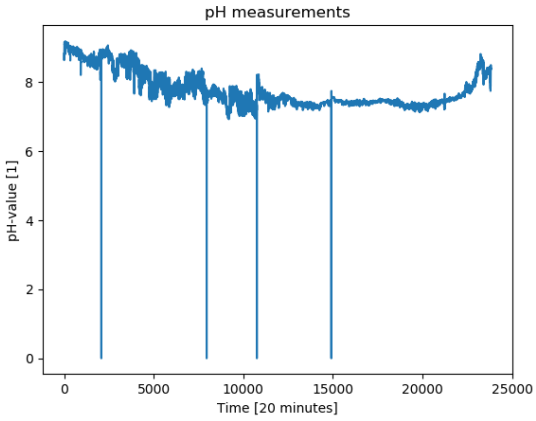


Fig. 13: Raw pH-value data collected by the sensor during a time period of 365 days.

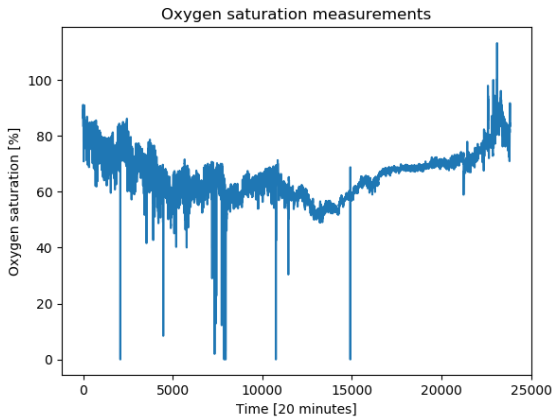


Fig. 16: Raw oxygen saturation data collected by the sensor during a time period of 365 days.

C. Data generation and simulation of sensors

FL requires multiple clients and since we only have had access to one sensor, we have had to virtually simulate clients. Clients were simulated by down-sampling from the data gathered by the only existing sensor, and then distributing it to K simulated clients.

TABLE II: Implementation parameters

Implementation parameters	Value
LSTM-cell-neurons of each LSTM-layer	50
Local batch size B	6
Loss function	Cross-entropy
Optimizer	Adam
Learning rate η	0.0005
L1-regularization-weight	0.0002
Number of clients K	5
Fraction of clients used in each round C	1

The physical sensor has a sampling period of 20 minutes, which means it makes a new measurement every 20 minutes, and uses a time window of 365 days. The data gathered by the sensor is distributed such that each simulated sensor gets a sampling period of 20·K minutes. If we for example simulate three sensors, every simulated sensor will get a new measurement every 60 minutes and will start gathering data at different times.

In Figures 11-16 we show the raw data measured by the sensor. The spikes in all figures occur as the sensor was malfunctioning at times. There has also been problems with algae being stuck on the sensor, which has caused issues with several measurements. More details on how this has affected our results will be brought up in the discussion.

D. Implementation parameters

The parameters used in our local machine learning models and in the FedAvg algorithm are shown in Table II.

When producing results we have been using 5 clients, which is around the realistic amount to be bought and deployed. The L1-regularization value of 0.0002 was used in the already developed LSTM-network, which worked well, and has therefore not been changed. The AdamOptimizer was used as optimizer. We settled for the learning rate $\eta = 0.0005$ since this value provided high accuracy without training taking too long. The centralized machine learning model had a batch-size of 30 training examples. Since we split up the data gathered by the sensor to 5 different clients, we set the local batch-size B to 6, as every client had access to one fifth of the data. The fraction of clients used in each global round C, was set to one. Lowering this fraction would reduce computing costs. However we did not want one more hyperparameter to tweak, and computing costs was not an issue. Both one and two LSTM-layers have been used.

E. Measuring accuracy and producing results

The global accuracy for the FL model was supposed to be measured as the total number of correct predictions divided by the total number of predictions made by each local model. In our case we decided to measure the global accuracy as the average local accuracy of all the clients. This is a good estimation of the global accuracy since all the clients have access to virtually the same amount of data, it only differs by maximum one training example. 10% of the data is used for testing and the rest for training.

TABLE III: Accuracy of the federated algorithm vs a centralized model for all measurement types

Measurement type	Global accuracy	Centralized accuracy
Salinity (SA)	0.98170	0.98816
Conductivity (CN)	0.92052	0.94123
Reduction potential (RX)	0.94079	0.97124
pH-value (PH)	0.76209	0.87489
Oxygen saturation (OS)	0.73823	0.87194

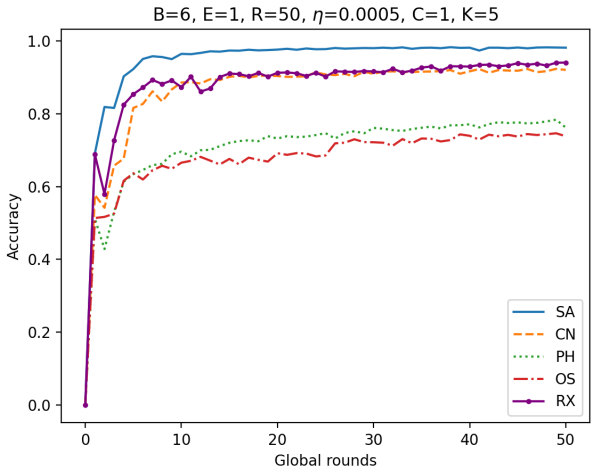


Fig. 17: Global accuracy, with one local epoch E and 50 global rounds R, of each measurement type plotted against number of global rounds R. The final values are shown in Table III.

TABLE IV: Global and centralized accuracy for input CN with different amount of total epochs, and the ratio between them

E/R/CE	Global accuracy	Centralized accuracy	CA ÷ GA
1/50/50	0.92052	0.94912	1.03107
5/50/250	0.93433	0.95332	1.02032
10/50/500	0.93562	0.95332	1.01892

When comparing the global and centralized accuracy, we decided to use the same number of total epochs in both cases. For example, if we use 250 epochs in the centralized training then $R \cdot E = 250$, where R denotes the number of global rounds and E denotes the number of local epochs, in the FL training.

We have also decided to use the measurements of CN in most of our experiments since the gathered data seemed relatively regular compared to other measurement types.

IV. Results

In Figure 17, the global accuracy is plotted against the number of global rounds for every available measurement type, when E=1 and R=50. The final global accuracy for each measurement type is provided in Table III as well as the final accuracy of a similar centralized model. The centralized model is 7.8% more accurate on average for the different measurement types than the federated model.

In Figure 18, the global accuracy is plotted against the number of global rounds when using one and two LSTM-layers and CN as measurement type, when E=1 and R=50.

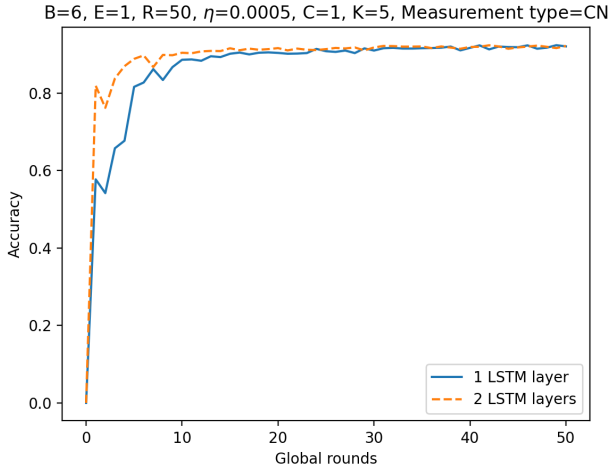


Fig. 18: Comparison of performance using one and two LSTM-layers.

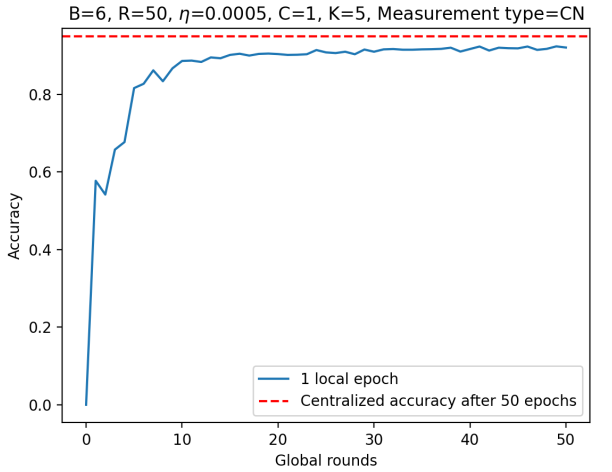


Fig. 19: Global accuracy with one local epoch and 50 global rounds plotted together with final accuracy of centralized model after 50 epochs. The final values are shown in the first row of Table IV.

In Figure 19-21 the global accuracy has been plotted against the number of global rounds for one, 5 and 10 local epochs E, respectively. The final global accuracy for each figure is provided in Table IV as well as the final accuracy of a similar centralized model for each case. In Table IV, CE denotes the number of epochs used in the centralized model. CA and GA are used as abbreviations for centralized accuracy and global accuracy, respectively.

CN is the measurement type used when producing the results presented in Figure 19-21 and Table IV.

V. Discussion

The results presented in Figure 17 show how different measurement types have reached significantly different accuracy. There are some obvious reasons to why this has happened and all of them are related to the raw data gathered by the sensor.

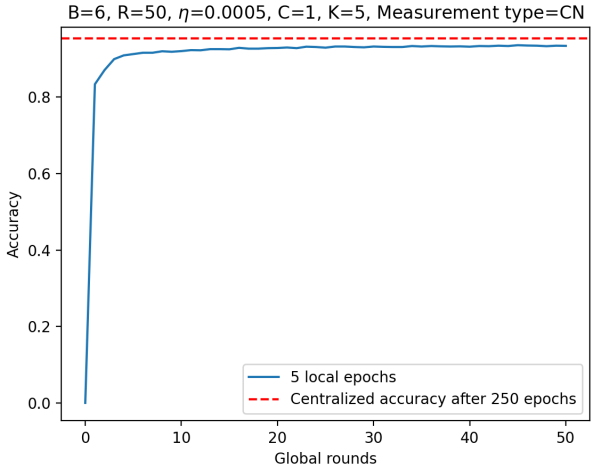


Fig. 20: Global accuracy with 5 local epochs and 50 global rounds plotted together with final accuracy of centralized model after 250 epochs. The final values are shown in the second row of Table IV.

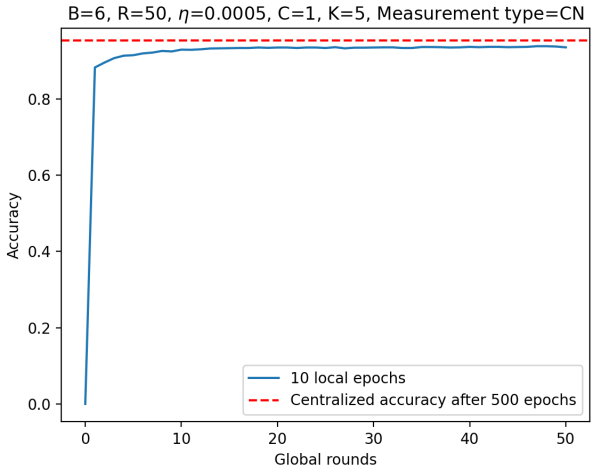


Fig. 21: Global accuracy with 10 local epochs and 50 global rounds plotted together with final accuracy of centralized model after 500 epochs. The final values are shown in the third row of Table IV.

If we look back to Figures 11-16 we realize that the measurements of OS and PH are the most volatile. This makes it harder for our machine learning model to predict them. The raw data also indicates that the measurement type most affected by sensor malfunctions and the algae issue discussed earlier, is RX. Paradoxically, this is one of the measurement types on which the FL model has attained the highest accuracy. This does however seem reasonable, since during the time the sensor has been malfunctioning, the fluctuation of the reduction potential data has decreased, which makes it more predictable.

The downwards spikes that can be observed in the raw data presented in Figures 11-16 originates from the sensor being completely shut down. Since this has not happened more than

five times over the course of the last year, which is the time window used for collecting data, we do not think that the impact of the sensor being shut down has had a significant effect on the prediction results.

The results presented in Table III show that the centralized model outperforms the federated model for all the measurement types. On average, the centralized model has a 7.8 % better accuracy than the federated model. However, this does not mean that the federated model produced in this project is useless. It is important to realize, as discussed in section III-C, that the amount of data available to train the simulated sensors has been reduced, which has led to the clients only getting access to one fifth of the data that they would have had access to if they were physical sensors. With this in mind, the results presented in Table III should be seen as promising, and as an incentive to conduct more research regarding deploying more physical sensors.

The results presented in Figure 18 indicates that faster convergence is achieved when using two rather than one LSTM-layer, however, the training time increases significantly as more weights and biases have to be tweaked.

The results presented in Figures 19-21 and Table IV seem to indicate that the global accuracy tends towards the centralized accuracy as the number of local epochs increase. If we would increase the number of local epochs and/or global rounds it could affect the global accuracy in three possible ways:

- It never reaches the centralized accuracy;
- It converges to the same accuracy as the centralized accuracy;
- It surpasses the centralized accuracy.

It would have been interesting to experiment with more local epochs and/or global rounds, however, this was too time consuming for the scope of this project.

VI. Conclusion

In this project we have implemented a FL algorithm with the intentions of detecting anomalies in measurements made by a water monitoring IoT-sensor.

Depending on the different measurement types, we reached an accuracy in the range of 73.8 % to 98.2 %. These results can be compared with a, previously developed, centralized machine learning model which achieved an accuracy in the range of 87.2 % to 98.8 %. On average, the centralized model had a 7.8 % better accuracy than the federated model. However, these results do not necessarily mean that the use of FL to improve anomaly-detection should be discarded, as simulating sensors have significantly reduced the training data for each simulated sensor. Problems with the deployed sensor, for example algae being stuck on it, have also had a negative impact on the results.

In summary, we believe that FL has the potential to produce interesting results, provided that more physical sensors are deployed.

VII. Future work

There are several aspects of our project that could be expanded upon, experimenting with different number of bins being one of them. It would also have been interesting to remove outliers from data in order to explore if better accuracy could be achieved. Another activity worthy of further investigation would have been to experiment with using more local epochs and/or global rounds in order to explore if the federated model converges to, or even outperforms, the accuracy of the centralized model.

Since most of the unsatisfying results were related to problems with the sensor, it would be reasonable to conduct further research with fully functioning equipment. This could be achieved either by more frequent cleaning and better maintenance of the sensor, or by investing in a better sensor.

Another interesting extension of the problem would be to compare a FL approach with a multi-client centralized approach. A multi-client centralized approach would imply uploading all the data from all the sensors to a main server, execute training on the server and then download the final global model to all clients for predictions. In order to conduct this comparison, additional physical sensors have to be deployed.

Acknowledgment

We would like to thank our supervisor José Mairton Barros da Silva Júnior for his invaluable support and guidance during the project.

References

[1] M. Gusev and S. Dustdar, "Going back to the roots—the evolution of edge computing, an iot perspective," *IEEE Internet Computing*, vol. 22, no. 2, pp. 5–15, 2018.

[2] H. Zheng, H. Hu, and Z. Han, "Preserving user privacy for machine learning: Local differential privacy or federated machine learning?" *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 5–14, 2020.

[3] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, May. 2020.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Singh and J. Zhu, Eds., vol. 54. Fort Lauderdale, FL, USA: PMLR, 20–22 Apr 2017, pp. 1273–1282. [Online]. Available: http://proceedings.mlr.press/v54/mcmahan17a.html

[5] A. Andersson. (2021, Apr.) iwater: Digital overvakning av stadens vattenkvalitet. [Online]. Available: http://miljobarometern.stockholm.se/vatten/samarbeten-och-projekt/iwater-digital-overvakning-av-stadens-vattenkvalitet/

[6] "Directive 2000/60/ec of the european parliament and of the council of 23 october 2000 establishing a framework for community action in the field of water policy," *Official Journal*, Sep. 2014. [Online]. Available: http://data.europa.eu/eli/dir/2000/60/2014-11-20

[7] G. Destouni, I. Fischer, and C. Prieto, "Water quality and ecosystem management: Data-driven reality check of effects in streams and lakes," *Water Resources Research*, vol. 53, no. 8, pp. 6395–6406, 2017. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2016WR019954

[8] J. Salonsaari. (2021, Apr.) Projektsbeskrivningsmall iot for innovativ samhallsutveckling, utlysning hosten 2017- varen 2018. [Online]. Available: https://www.digitaldemostockholm.com/media/1037/iwater-vinnova-approved-1.pdf

[9] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.

[10] F. Rosenblatt, "The perceptron - a perceiving and recognizing automaton." Report 85-460-1, Cornell Aeronautical Laboratory, 1957.

[11] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition*. Sebastopol, California: O'Reilly Media, Inc., Sep. 2019, ch. 15.

[12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[13] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2017.
[14] X. Huang, H. Tan, G. Lin, and Y. Tian, “A lstm-based bidirectional translation model for optimizing rare words and terminologies,” in *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*, 2018, pp. 185–189.
[15] S. An, Z. Ling, and L. Dai, “Emotional statistical parametric speech synthesis using lstm-rnns,” in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2017, pp. 1613–1616.
[16] L. Huang and C.-M. Pun, “Audio replay spoof attack detection using segment-based hybrid feature and densenet-lstm network,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 2567–2571.
[17] I. C. Education. (2021, Apr.) Recurrent neural networks. [Online]. Available: https://www.ibm.com/cloud/learn/recurrent-neural-networks
[18] M. Wallén and M. Böckin. (2021, May.). [Online]. Available: https://github.com/mauriciobvaldes/FedAvg_iWater

Machine Learning for Water Monitoring Systems

Robirt Asaad and Carlos Ribé

***Abstract*—Water monitoring is an essential process that manages the well-being of freshwater ecosystems. However, it is generally an inefficient process as most data collection is done manually. By combining wireless sensor technology and machine learning techniques, projects such as iWater aim to modernize current methods. The purpose of the iWater project is to develop a network of smart sensors capable of collecting and analyzing water quality-related data in real time.**

To contribute to this goal, a comparative study between the performance of a centralized machine learning algorithm that is currently used, and a distributed model based on a federated learning algorithm was done. The data used for training and testing both models was collected by a wireless sensor developed by the iWater project. The centralized algorithm was used as the basis for the developed distributed model. Due to lack of sensors, the distributed model was simulated by down-sampling and dividing the sensor data into six data sets representing an individual sensor. The results are similar for both models and the developed algorithm reaches an accuracy of 98.41 %.

***Sammanfattning*—Vattenövervakning är en nödvändig process för att få inblick i sötvattensekosystems välmående. Dessvärre är det en kostsam och tidskrävande process då insamling av data vanligen görs manuellt. Genom att kombinera trådlös sensorteknologi och maskininlärnings algoritmer strävar projekt som iWater mot att modernisera befintliga metoder.**

Syftet med iWater är att skapa ett nätverk av smarta sensorer som kan samla in och analysera vattenkvalitetsrelaterade data i realtid. För att bidra till projektmålet görs en jämförande studie mellan den prediktiva noggrannheten hos en centraliserad maskininlärningsalgoritm, som i nuläget används, och en distribuerad modell baserad på federerat lärande. Data som används för träning och testning av båda modellerna samlades in genom en trådlös sensor utvecklad inom iWater-projektet. Den centraliserade algoritmen användes som grund för den utvecklade distribuerade modellen. På grund av brist på sensorer simulerades den distribuerade modellen genom nedprovtagning och uppdelning av data i sex datamängder som representerar enskilda sensorer. Resultaten för båda modellerna var liknande och den utvecklade algoritmen har en noggrannhet på 98.41 %

***Index Terms*—Federated learning, Internet of Things, Decentralised data, Distributed learning, Long Short-Term Memory.**

Supervisors: José Mairton Barros Da Silva Júnior and Carlo Fischione

TRITA number: TRITA-EECS-EX-2021:180

I. Introduction

A. Background

The impact freshwater has on our lives is difficult to overstate as its necessity permeates nearly all aspects of our existence. Not only are ecosystems such as lakes, rivers and groundwater basins mankind's main source of drinking water, they are also essential for most pillars of society including power production, agriculture, industry, and transportation [1]. Unfortunately, freshwater is a finite resource which only makes up about 3% of all water on earth [2], and as the population continues to grow, so does the need for access to such sources [2]. Despite their importance, and due to issues such as pollution and global warming, freshwater ecosystems are among the most threatened [1].

To combat their deterioration, water quality is monitored as it plays a pivotal role on the health of these ecosystems [1]. The process of monitoring water quality generally begins by manually gathering test samples from a freshwater source. The samples are then transported to laboratories where they are analyzed. These laboratories however, are normally located great distances away from where the samples were originally collected [2], meaning the entire process is often costly and time-consuming. However, due to the rapid increase of both quality and quantity of sensor and communication technology, more efficient methods are currently being developed around the globe to help make water quality monitoring more effective.

iWater, a part of the strategic innovation program for the Internet of Things in Sweden (IoT Sweden), aims to develop and install a water monitoring network of wireless sensors capable of automatically gathering data indicative of water quality. To achieve this, the collected data is wirelessly transmitted to a central server where further analysis is done by a machine learning (ML) program with the purpose of finding anomalies in the water. However, this innovative method presents its own set of challenges.

One such challenge is that these sensors often are set in places far away from reliable internet connections and may have very limited bandwidth. Another issue is that unforeseen characteristics in the data as well as large variations may give false positives and false negatives. A possible solution to these problems may be a rather new ML paradigm introduced in [3] which was released by Google in 2016. The technique described in this paper is referred to as federated learning (FL) and it provides an alternative to traditional methods of ML, where data collected by edge devices (such as mobile phones or in our case sensors) is sent back to central servers to train a common ML model [3]. FL allows devices to train their own models locally and only send training parameters to a central server, where these parameters are aggregated and sent back to all devices in the network to further improve training [3], see Figure 1.

B. Problem

To contribute to the iWater project, the project group investigated if FL could be a suitable method of analyzing data, collected by a network of similar sensors. To achieve this, we simulated a ML network based on a FL algorithm known as Federated Averaging (FedAvg). Then compared its performance to that of the (centralized) model currently

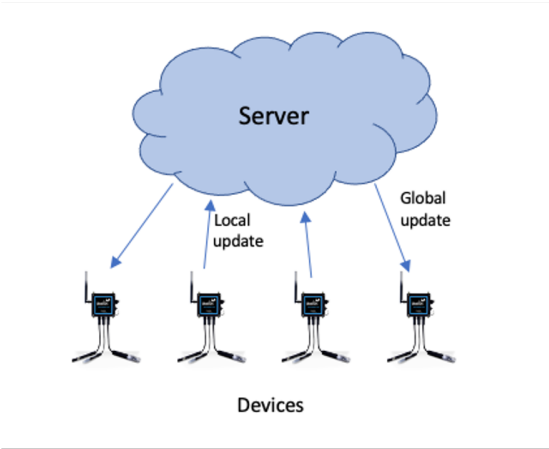


Fig. 1. A network of edge devices connected to a central server. The devices send their training parameters to the server where these parameters are aggregated and sent back to improve the next training round.

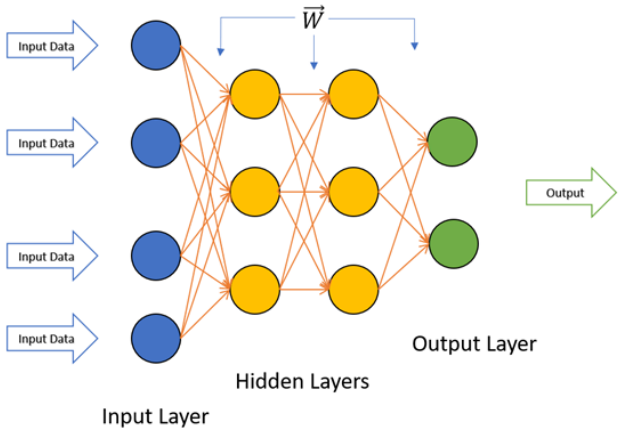


Fig. 2. ANN with layers of neurons connected trough channels with assigned values called. weights. Input gets into the system via the input layer. Most calculations are then done in the hidden layers. The output layer holds the set of possible solutions the network is trained to output.

used to analyze data collected by the iWater sensor. Both models are trained on an input $In = T + X$ such that $X \in \{PH, CN, SA, OS, RX\}$, where T is the water temperature and X is the value of a chemical or biological property in the water (See III-A for further information). The output is the prediction of the approximate value of X collected in the next sample.

II. Theory

This section contains descriptions of the main concepts and algorithms needed to understand and actualize the ML models tested in this project. First, the concept of Deep learning more specific Artificial Neural Networks and supervised learning are introduced, followed by a summary of how Long Short-Term Memory networks function. The concept behind FL, as well as the specific algorithm FedAvg are described towards the end.

A. Artificial Neural Networks

ML algorithms are computer programs that can be trained to make predictions by learning to recognize patterns in data. Artificial Neural Networks (ANN) are a type of ML. ANNs are generally taught to learn patterns by first feeding them hundreds or thousands of data instances and later testing them on novel data to analyze and improve their performance. There are different ways to train ANNs. The method used to train the neural networks used for this project is called *supervised learning*. It is a method where every data instance used for training is marked with a label.

By training an ANN with labeled data, the algorithm can eventually learn patterns and output predictions on new, unlabeled data. Item classification or the prediction of a numerical value are two common tasks in supervised learning. In tasks concerning item classification, an ANN may be trained with examples of a certain class of items, and learn to predict new value. Likewise, if an ANN is tasked to predict a numerical value, it may be trained with a set of features that are one way or another associated with those values. Both methods are based on a regression model [4] [5].

A simple ANN (see Figure 2) normally consists of an input layer where the training data is entered to the network, a number of hidden layers, where most of the data processing is done, and an output layer. The output layer normally holds the set of possible outputs the network can choose have. Each layer is made of a so-called artificial neuron and if every neuron in a layer is connected to every neuron in the previous layer it is called dense layer. Each connection is given a numerical value called a *weight*, and each neuron has an activation value based on the weighted sum of previous layer. The weighed sum is added to a bias which tells what the minimum activation value the neurons should have,

$$(w_1x_1 + ... + w_mx_m) + bias = \sum_{i=1}^{m}(w_ix_i) + bias, \qquad (1)$$

where w_i is the weight, x_i is the input data, $i = 1, ..., m$ and m is number of neurons. Then applies an activation function φ to the weighted sum and the bias to ensure that the value out from the neuron is in the range of 0 and 1, which can describe how active the neuron is [5] [6]. The output y of one neuron is thus

$$y = \varphi(\sum_{i=1}^{m} w_ix_i + bias). \qquad (2)$$

The more often a neuron is activated by a neuron from a previous layer, the higher the value of the weight between them gets. The algorithms used for this study are based on a class of ML algorithms known as *Deep Learning*. Deep learning algorithms are generally made of vast sets of ANNs. When data propagates from the input layer through the network, such a network is referred to as a *Feedforward Neural Network* [4] [5]. In our algorithm we use Softmax activation function on the output layer, which is based for a logistic regression model

to use in multiple classes. The activation function is

$$\hat{p}_a = \frac{\exp(x_a)}{\sum_{j=1}^{A} \exp(x_j)}, \tag{3}$$

where $\hat{p}_a$ is the estimated probability and x_a is the example for class a, and A is number of classes [5].

To estimate how well the algorithm predicts, a loss function is applied to the output of the network, which can be written as $f(x) = l(x_i, y_i; w)$ where (x_i, y_i) is input-output data and w are the weights. The loss function evaluates the difference between the expected output and the actual output and this is called the output error.

The loss function uses a method called *cross entropy* because it is especially good for a logistic regression model. It measures the difference between two probability distributions; a target probability y and an estimated probability $\hat{p}$. If the algorithm predicts a high probability value the model gets rewarded but if it predicts a low value, it instead penalizes the model. This way, the algorithm will know how well it performance [5]. The cross entropy with class a is

$$H(y_a, \hat{p}_a) = -\sum_{a=1}^{A} y(x)_a \log \hat{p}_a, \tag{4}$$

To optimize the output error we minimize the loss function

$$\min_{w \in \mathbb{R}^d} f(x) \quad \text{where} \quad f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(w), \tag{5}$$

and n is the total number of data samples. The loss function is minimized by calculating its gradient $\nabla f_i(w)$ and then applying back-propagation. With back-propagation, the gradient propagates backwards in the network to update all weights. This method can be further optimized by taking small steps until either the minimal value of the function is reached or a predefined number of iterations (epochs) have passed. This is also called Gradient Descent and its step size is decided by a parameter called the *learning rate* η [5].

For optimization we used adaptive moment estimation (Adam). An Adam optimizer is based on adaptive learning rate, which uses step estimation to adapt the learning rate for each weight in the network [7]. To avoid overfitting, a function called *Least Absolute Shrinkage and Selection Operator Regression* (Lasso Regression or $\ell 1$ Regularization) was used. Lasso Regression adds a term (weights) to the loss function and the advantage is that it reduces the value of weights of less important features to zero [5].

B. Long Short-Term Memory Networks

LSTM networks consist of Recurrent Neural Network cells. LSTM networks are trained on data sent sequentially into a layer, where each cell in the layer simultaneously receives an input and output from a previous cell. This allows an LSTM network to identify which information is more important to retain and which should be discarded. One advantage these networks have, is that they transport data in long sequences and are therefore good at holding information. In other words, they can function as memory cells. Another advantage is that they do not suffer from vanishing gradient problems, which arise when the gradient of a function becomes so small that it no longer has any effect in the training of a ML algorithm [5] [8].

C. Federated Learning

FL is a rather new approach to ML developed by Google [3]. It was created to collaboratively train shared prediction models across networks of devices, while simultaneously keeping all training data on the individual devices that gathered the data [3]. FL is best suited for problems that have properties such as 1) Training on real-world data collected from from devices that provide an advantage over traditional training on data that is generally available on servers, 2) Training on data that is sensitive and private 3) For supervised ML tasks with labeled data can be easily inferred from user interaction [3].

Client is connected to a distributed network through a common server. The idea is to train clients locally with their own local dataset. Then the trained local model is sent to the server which makes an average from all local models of all devices and sends the new updated global model to the devices, then this process is repeated for each global round [9]. The loss function can then be expressed according to

$$f(x) = \sum_{i=1}^{K} \frac{n_k}{n} F_k(w) \quad \text{and} \quad F_k(w) = \frac{1}{n_k} \sum_{i \in P_k} f_i(w) \tag{6}$$

where a client k has the dataset P_k, which gives $n_k = |P_k|$ and n is the total dataset over all clients [3].

D. Federated Averaging

In the FedAvg algorithm (see Figure 3) each device train their individual ML model on local data by calculating the gradient of F_k for w_t in global round t. When the local training is complete, the clients' training parameters w_t^k are sent to a central server where they get aggregated with other training parameters from other sensors, see equation 7. After the aggregation is finished, the aggregated parameters are sent to update all devices and then the cycle repeats [3].

$$w_{t+1} \leftarrow \sum_{i=1}^{K} \frac{n_k}{n} w_{t+1}^k \tag{7}$$

The algorithm used for the FL model is shown in Figure 3. It starts by initializing the weights w_0 for the global model [3]. Then the first FedAvg round starts with the server randomly selecting a set of m clients S_t, with the maximum number of clients per round according to $C * K \geq 1$, where C is the fraction of clients and K is the total number of clients. Then w_0 is sent to the selected clients. The clients then train w_0. After receiving the updated weights, they are trained with local data that has been scaled down to mini-batches B for each epoch round E. This gives the new local weights w_{t+1}^k for client k. The next step is to update all the weights w_{t+1}^k from the clients to the global model that averages the weights.

The last step is sending the new averaged weights to the clients to train them locally. This process continues in t

number of global rounds [3].

Algorithm 1: Federated Averaging

Server Update:
Initialize w_0
for each round t = 0,1,... **do**
$m \leftarrow max(C * K, 1)$
$S_t \leftarrow (random\ set\ of\ m\ clients)$
for each client $k \in S_t$ **in parallel do**
$w_{t+1}^k \leftarrow ClientUpdate(k, w_t)$
$w_{t+1} \leftarrow \sum_{i=1}^{K} \frac{n_k}{n} w_{t+1}^k$

ClientUpdate(k, w)**:**
$B \leftarrow (split\ P_k\ into\ batches\ of\ size\ B)$
for each local epoch i from 1 to E **do**
for batch $b \in B$ **do**
$w \leftarrow w - \eta \nabla l(w; b)$
return w to server

Fig. 3. Algorithm for FedAvg. $w_0$represents the weights and biases of the network. C is the fraction of clients used to train the algorithm. m is the total number of clients in the subset S_t is total number of clients, E is number of local epochs, and B the number of local batches [3]

III. Method

Both the centralized algorithm as well as the distributed model created for the study were written in Python. The centralized algorithm was handed to us by our supervisors and functioned well so no changes to it were made. The same algorithm was used as the basis for the simulated individual ML models used for the distributed network. It was implemented using Google's Python library, `Tensorflow 1.13`.

To handle the data more easily, the libraries `Numpy` and `Pandas` were used. The simulated model was done mostly using standard python libraries, as well as with `Numpy` and `Pandas`. The plots used in the report were done with the help of the `matplotlib` library. A full version of the code can be found at [10].

A. Data collection

The raw data used for the project was collected by a water monitoring sensor and sent wirelessly to a server. The sensor collected different types and six of them (see Figures 4-9) were used for training and testing the algorithms. The sensor takes samples once every 20 minutes and datasets collected over a period of 365 days were used to train and test the models. The centralized model trained on one entire dataset at a time. In contrast, to simulate the network of sensors for the distributed variant, a dataset had to be down-sampled and divided into six parts. Each part represented an individual virtual sensor. A dataset was divided such that the first virtual sensor received the first data sample, the second received the next data sample and so on.

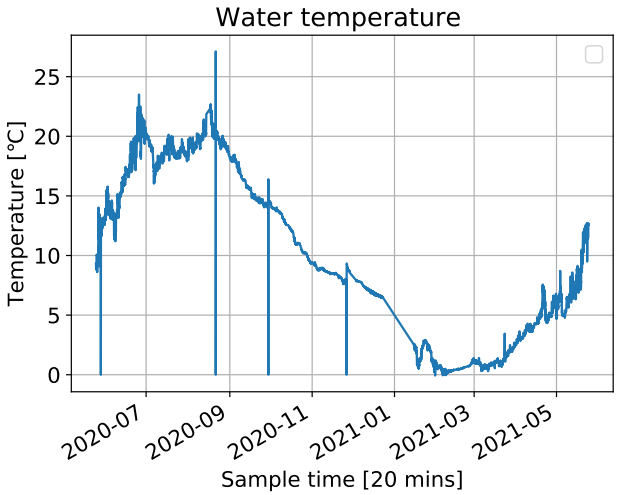


Fig. 4. Water Temperature (T) measurements collected once every 20 minutes during a time window of 365 days. Changes in this parameter greatly affects other water quality data. It is always used as part of the input because of this.

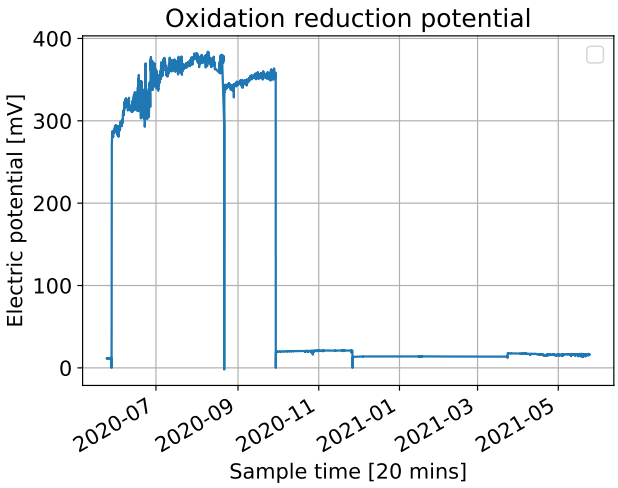


Fig. 5. Oxidation Reduction potential (RX) measurements collected once every 20 minutes during a time window of 365 days.

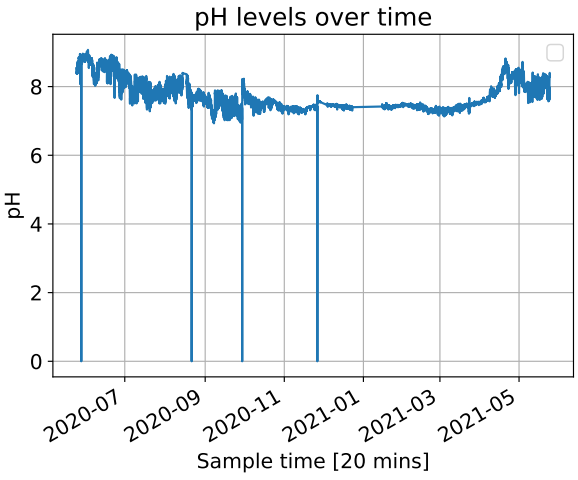


Fig. 6. pH (PH) measurements collected once every 20 minutes during a time window of 365 days.

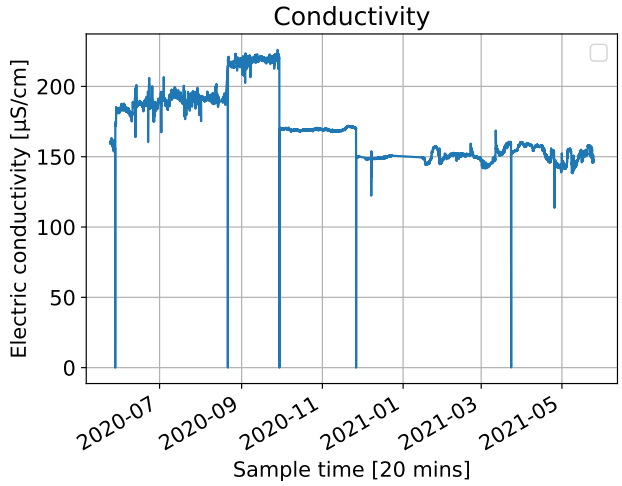


Fig. 7. Water Conductivity (CN) measurements collected once every 20 minutes during a time window of 365 days.

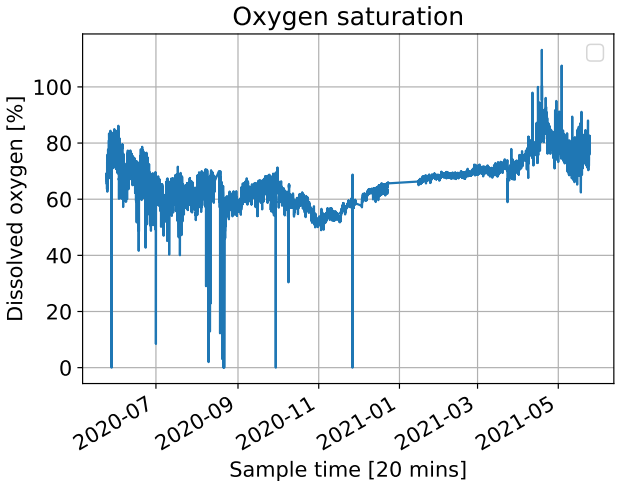


Fig. 9. Oxygen Saturation (OS) measurements collected once every 20 minutes during a time window of 365 days.

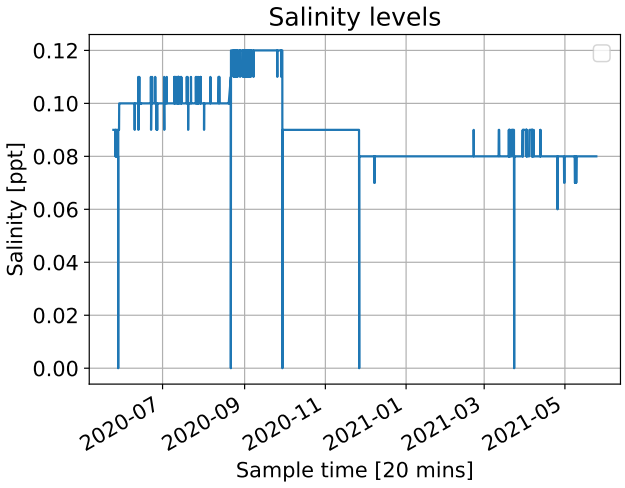


Fig. 8. Salinity (SA) measurements collected once every 20 minutes during a time window of 365 days.

B. Data preprocessing

Firstly, the dataset was downloaded using MongoDB, in the form of a list with two columns; the first column held the values of each individual sample and the second contained the times in which they were collected. The timestamped data was then divided into a number of bins such that the numbers of the raw data in each bin were the same size (This step is depicted in Figures 2 & 3 as the layer of the local model).

The size of each bin is based on the percentage of distributed data that ends up in it [11]. The number of bins was used as a testing parameter and varied between six and ten. This was done using a method called ”qcut” from the Pandas library.

To better fit the models, the data was normalized and reshaped further. We ask the reader to visit [10] where the entire code is freely accessible. 90% of the data is used to training the algorithm and 10% of the data is used to test the algorithm on how good it is predicting.

C. Local Model

The centralized model was also used as the basis of our local model. It consists of an ANN with three to four different layers (see Figures 2 & 3), an input layer, an output layer (dense), and one or two hidden layers (LSTM) [4] [5]. The number of LSTM layers was used as one of the parameters for testing and comparisons between one and two layers were made. The LSTM layer’s outputs are connected to a dense layer, whose output is a probability function which predicts the value of the sample.

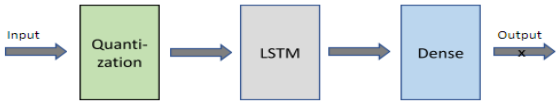


Fig. 10. Block diagram of the local model with one LSTM layer.

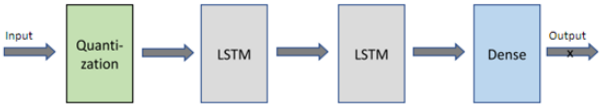


Fig. 11. Block diagram of the local model with two LSTM layers.

D. Federated Averaging

The simulated network consisted of only six sensors. Because of this, the FedAvg algorithm described in section II-D was tweaked as follows. Instead of choosing a random set of

clients S_t, each global round trained the entire set of all six sensors. Another difference was that, instead of initializing the weights and biases globally using the same initial values for all sensors, the variables were initialized locally. The clients in our case are the simulated sensors and the training data is data that has been collected from the one existing sensor. To simulate several sensors (the clients), data was collected from a server and then split into K different segments.

E. Test parameters

Different parameters were chosen as variables for the purpose of testing and comparing both ML models in different ways. Besides comparisons between different data types, The main data type used during most of the testing was salinity (SA) due to the uniformity of the values in the dataset. The test parameters can be seen in Table 1.

TABLE I
TEST PARAMETERS USED FOR THE SAKE OF COMPARISON BETWEEN THE CENTRALIZED AND FL MODELS.

Parameter	Label	Values
Epochs	E	1,5,10,20
Number of Bins	B	6,8,10
Number of LSTM layers	L	1,2

IV. Results

Figures 12 and 13 show the varying results while using different types of training data described in III-A. The accuracy of both models show similar patterns. The highest accuracy values were reached when using salinity as input data. Using oxygen saturation as training data yielded the lowest accuracy levels in both models. The accuracy of the centralized model converged towards the same value when using conductivity as well as oxygen reduction potential as training data. Using the same sets of data in the federated model however, the accuracy converged toward different values.

When training the local model for each data type and comparing the centralized model with the distributed simulation, the final test results of every individual sensor were saved after every training round.
The values were then averaged together and the 50 resulting values where used to plot all of the distributed model's graphs, see Figure 12 to Figure 17.

The accuracy comparison of the federated model for varying number of epochs is shown in Figure 14. The model used one LSTM layer and the number of bins was set to six. Figure 15 depicts a comparison between the federated and centralized model using different numbers of bins and LSTM hidden layers. The model was set to train on five epochs per global round. In Figure 16, the accuracy comparison between federated and centralized model with five epochs respective 50 epochs. Both models use one LSTM and eight bins. Finally, Figure 17 shows the accuracy comparison the accuracy comparison between federated and centralized model with five epochs respective 50 epochs. Both models use two LSTM and eight bins.

TABLE II
ACCURACY COMPARISON BETWEEN THE CENTRALIZED MODEL AND THE FL ALGORITHM.

Data type	Cetralized acc (%)	FL acc (%)
SA	96.720	98.067
CN	94.575	91.667
RX	94.617	94,545
PH	86.543	75,902
OS	86.585	75.816

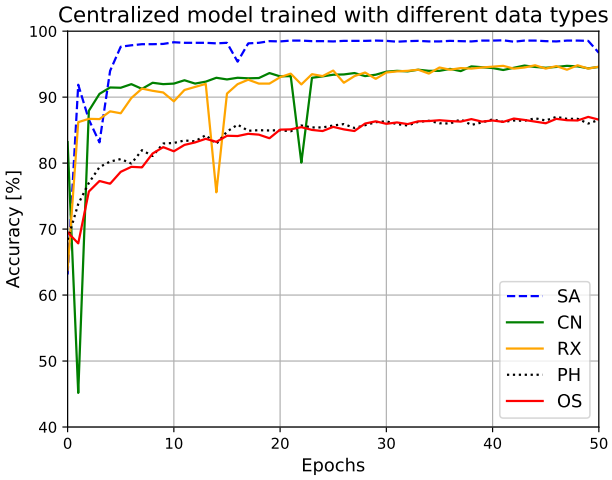


Fig. 12. Accuracy levels of the centralized model trained on 5 different data types. Salinity (SA), water conductivity (CN), Oxygen reduction potential (RX), pH value (PH), Oxygen saturation (OS).
Test parameters: E = 50, B = 6, L = 1.

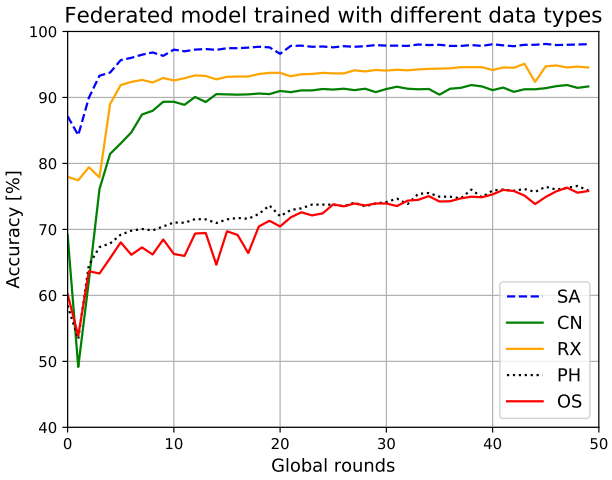


Fig. 13. Accuracy levels of the federated model trained on 5 different data types. Salinity (SA), water conductivity (CN), Oxygen reduction potential (RX), pH value (PH), Oxygen saturation (OS).
Test parameters: E = 5, B = 6, L = 1.

V. Discussion

The results show how the model's accuracy drastically fluctuates depending on the type of input data selected for training. Results with the highest accuracy were achieved when using salinity levels as input for both models (Figures 12 and

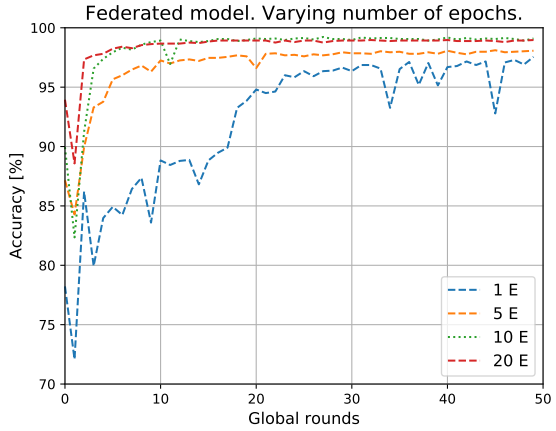


Fig. 14. Accuracy comparison of the federated model between varying number of epochs. Testing parameters: E = variable, B = 6, L = 1.

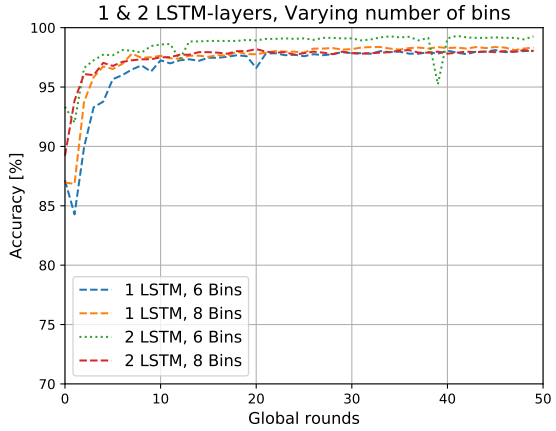


Fig. 15. Accuracy comparison of the federated model with varying number of both bins and LSTM layers.
Test parameters: E = 5, B = variable, L = variable.

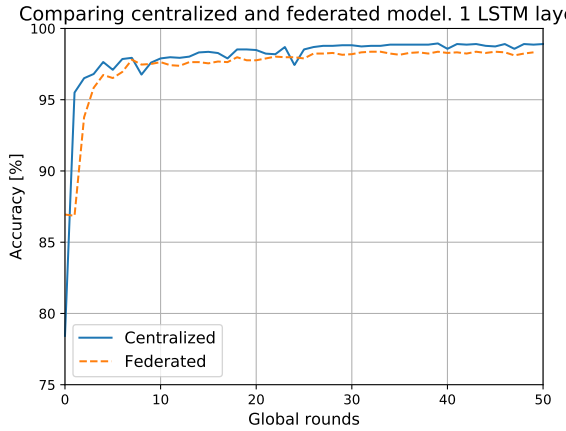


Fig. 16. Comparison between the centralized and the federated model using one LSTM layer. Test parameters: Centralized: E = 50, B = 8, L = 1.
Federated: E=5, B = 8, L = 1.

13). This was probably because the values in that dataset

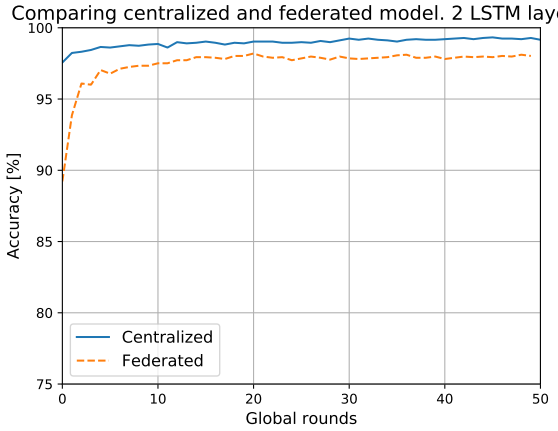


Fig. 17. Comparison between the centralized and the federated model using 2 LSTM layers. Test parameters: Centralized: E = 50, B = 8, L = 2.
Federated: E=5, B = 8, L = 2.

do not vary as much as with other data types. In contrast, oxygen saturation gave the lowest accuracy overall and Figure 9, shows that it was the dataset with the highest variation. The number of epochs appears to be an important variable for the accuracy of the models; at least up to a certain level.

Figure 14 shows that when the number of epochs increased, the accuracy did as well. However, accuracy levels converged toward almost the same value when testing both 10 and 20 epochs. Given the graphs showed in Figure 15, the number of bins was not as important as other training parameters, as three of the four graphs converged toward the same value. However, there was a distinct improvement in accuracy when using six bins and two LSTM layers. Figure 15 also show what might be a result of over- training when using six bins and two LSTM layers. Figures 16 and 17 show that the number of LSTM layers does improve accuracy for both ML models.

VI. Conclusion

The results from this comparative study show that had accuracy of the distributed model was close to the accuracy of the centralized model. The distributed model reached managed to reach results with up to 98.41% accuracy which was slightly better than we expected and even though the centralized model was capable of reaching slightly higher accuracy, we believe that further research related to the simulated model should still be considered.

VII. Future Works

Simulating the federated network was a demanding task for our computers and we would therefore strongly recommend future researchers to either install TensorFlow 1 on the GPU directly or use TensorFlow 2, which automatically runs on the GPU and can decrease the run time by 85% [12]. Another option would be to use a multiprocessing module to reduce the time of running such a demanding program by running several cores simultaneously. The longer simulations we ran could could easily take around 13-14 hours and a few even close to 20 hours. Multi-threading is another alternative which

could help reduce the amount of time it takes to run the simulation. In multi-threading, the processor uses several threads simultaneously to preform more calculations and therefore it can reduce the simulation time [13].

As previously mentioned, an advantage of FL is that data can be stored locally to preserve privacy. However, there is a disadvantage when updating the local model to the server. During the update, it can leak sensitive information, for example when typing in a mobile phone, it can recognize a certain pattern and reveal one's password or credit number. Therefore, a more secure FL model should be looked into. As for testing, We would have liked to make further comparisons on different combinations of hyper-parameters. Further simulations should also test for different ways of splitting data more realistically. We also would have wanted to test how a version of the distributed model where trainable variables where initialized globally would compare to the one we used.

ACKNOWLEDGMENT

We want to thank our supervisor José Mairton Barros Da Silva Júnior for his guidance and support through the project.

REFERENCES

[1] J. Bartram, R. Ballance, WHO, and UNEP, *Water quality monitoring: a practical guide to the design and implementation of freshwater quality studies and monitoring programmes: published on behalf of United Nations Environment Programme World Health Organization*, 1st ed. E FN Spon, 1996.

[2] F. Jonas, *Drinking water: sources, sanitation and safeguarding*. Swedish Research Council Formas, 2009.

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, A. Singh and J. Zhu, Eds., vol. 54. PMLR, Apr. 2017, pp. 1273–1282. [Online]. Available: http://proceedings.mlr.press/v54/mcmahan17a.html

[4] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge: MIT Press, 2016, http://www.deeplearningbook.org.

[5] A. Géron, *Hands-on machine learning with Scikit-learn, Keras, and TensorFlow : concepts, tools, and techniques to build intelligent systems*, 2nd ed. O'Reilly, 2019.

[6] M. Nielsen. (2019, Dec.) Neural networks and deep learning. [Online]. Available: https://http://neuralnetworksanddeeplearning.com/chap1.html

[7] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[8] M. Phi. (2018, Sep.) Illustrated guide to lstm's and gru's: A step by step explanation. [Online]. Available: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

[9] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, May 2020.

[10] C. Ribé and R. Asaad. (2021, May) Ef112x – degree project in electrical engineering, first cycle. [Online]. Available: https://github.com/Carlossr-1208/EF112X-Degree-Project-in-Electrical-Engineering-First-Cycle

[11] C. Moffitt. (2019, Oct.) Binning data with pandas qcut and cut. [Online]. Available: https://pbpython.com/pandas-qcut-cut.html

[12] (2019, Oct.) Tensorflow 2 - cpu vs gpu performance comparison. [Online]. Available: https://datamadness.github.io/TensorFlow2-CPU-vs-GPU#:~:text=The%20Conclusion,required%20training%20time%20by%2085%25

[13] M. Martin, D. Sorin, H. Cain, M. Hill, and M. Lipasti, "Correctly implementing value prediction in microprocessors that support multithreading or multiprocessing," in *Proceedings. 34th ACM/IEEE International Symposium on Microarchitecture. MICRO-34*, 2001, pp. 328–337.

CONTEXT L – PART II
AIOT: ARTIFICIAL INTELLIGENCE AND THE INTERNET OF THINGS

POPULAR DESCRIPTION

You Might be Unknowingly Inviting Hackers Over Your Doorstep

How would you feel if a hacker was having a chat with your baby? In this day and age, all kinds of devices are connected to the internet, even your baby monitors. Security on these devices are weak and developers need to keep up with the attackers. Like a game of chess, cybersecurity is all about outsmarting the opponent.

Technology can be very exciting and we buy new devices without hesitation to keep up with the latest trends. In some cases this might be harmless, but in others it could have serious consequences. Internet-connected devices like your printer, baby monitor and smartphone are all examples of such devices. You would definitely not want any of them to be used to harm you or your family. In 2018, it was revealed that baby monitors had been used in several kidnapping attempts. In one case, a family in Texas woke up in the middle of the night to a voice coming from their newborn's room. Someone was talking to their child but no one was in the room. The baby monitor was being controlled by a hacker from a remote location.

This is why cybersecurity is important. Due to the complexity and connectivity of modern devices, it is a vast topic and there are many aspects to consider when designing a secure system. To only act as a defender is not enough; one also needs to consider the attacker's point-of-view in order to cover all vulnerabilities. Cybersecurity can be seen as an everlasting chess game, where the defender and attacker try to outwit each other. By playing both roles the developer of a device achieves a secure product.

The need for cybersecurity grows with the number of connected devices. Improving the security is therefore important to ensure individual integrity and a safe society in the future.

SUMMARY OF PROJECT RESULTS

As society becomes more digitized, the use of Internet of Things (IoT) devices has become more widespread. Due to this increase, exploitation of IoT-devices has become a serious issue. This has caused some of the world's largest Distributed Denial of Service attacks, where the goal is to make a website inoperational by overloading it. The largest example ever recorded happened in September 2016, when the malware Mirai was used to take control of insecure IoT devices, such as IP cameras, and carry out such an attack. Since IoT-devices have limited hardware resources and the market competition is high - resulting in cheap products - the security aspect is seldom prioritized. Because of this situation, cybersecurity is becoming more relevant and a device's ability to withstand attacks in any form is essential to improve consumer privacy. The project groups in this context have worked towards improving the cybersecurity in the field of IoT because the system as a whole is only as strong as its weakest link.

The groups in project L6 have evaluated the cybersecurity of IoT-devices. This involves threat-modeling the devices and practically testing for the most probable and severe vulnerabilities. The project groups evaluated the security of an IoT-3D printer and a home-security camera. The process included creating an architecture overview of the product components and dataflows. From these, potential attack-surfaces and entry-points to the systems were identified, which in turn were used to map common vulnerabilities in the IoT-products. They were ranked on probability and the most common penetration methods based on these vulnerabilities were performed in practice. The results of this process are security evaluations of the products which can indicate improvements. To expand the product evaluations based on the projects, additional domains of

the product could be tested and possible countermeasures to patch the vulnerabilities could be presented. Future work could include conducting the same tests on similar products but from other vendors.

The focus of project group L7 was to validate and expand on the enterpriseLang Meta-Attack Language. This Domain-Specific Language (DSL) is used to model threats against a specific target system by generating fully-fledged attack-graphs based on probability. These attack-graphs give the user a graphical overview of potential security risks. In the past, each subsystem would require a new specific attack-graph to figure out security flaws; utilizing DSLs eliminates this arduous method. The aim of this project was to increase the attack coverage of enterpriseLang to reach 100%, improving the DSL. Furthermore, new design guidelines were useful in suggesting changes to the language. One great benefit of DSLs is the ease of use for non-cybersecurity personnel. With this in mind, enterpriseLang can be used by developers who lack knowledge in the domain of cybersecurity, to improve the security of their products. As the amount of IoT-products is increasing every year, enterpriseLang can contribute a great deal to the security of aforementioned products. Future project groups should focus on the development of other Domain-Specific Languages by using the proposed guidelines and methods of generating attack coverage, as was done for enterpriseLang.

Project group L9 put under scrutiny a state-of-the-art continuous authentication (CA) system used to authenticate users passively on their devices. By recording the behavior of the user, CA frameworks generate a user score with some frequency and adjust the user privileges accordingly. This means that if the recorded behavior at any time is dissimilar to that of an authenticated user of the device, the device can prompt a password entry for the user to continue using the device, or restrict the access to sensitive features. In practice, it is unlikely that a human would be able to mimic the detailed behaviour of a peer to the point of fooling a state-of-the-art CA system. However, recent development in the field of generative modelling, which is the method of reproducing data with the same distribution as some original data by machine learning means, tends to suggest that a machine might be able to do so. To this end, a proposed CA solution was chosen to be attacked. The solution is based on biometric data in the form of swipe patterns using several machine learning models for classification, i.e., classifiers. Using a generative model, swipe patterns of the same distribution as for authenticated users were generated. The generated swipes were then fed to the classifiers with the aim to impersonate authenticated users. The experiment indicates the degree of similarity between the genuine samples and the forged samples. The direct extension of this project would be to examine a generative model's capability of evading a more complex CA system which relies on a combination of different biometric data. The consensus is that such an authentication system would be more robust against attacks in general, which seems to suggest that it would be a greater challenge for generative models.

IMPACT ON SOCIETY AND ENVIRONMENT

Society strives for efficiency, this leads to constant optimization of its societal functions. The individuals in turn rely on the societal functions for their everyday life and play a role in maintaining this infrastructure. Moreover, every person seeks efficiency in their own lives. A modern approach to these strives is digitalization. For example, the transition from wired phones to mobile phones, and then smartphones, has allowed individuals to be more flexible in arranging meetings and reaching loved ones. On a societal level, the greater connectivity achieved by the transition has improved the distribution of information. A general aim is that the strive for efficiency should be in line with sustainable development.

As more devices get connected to the internet, the concern for privacy and security becomes a pressing matter for all parts of society, from individual consumers to infrastructure levels. We saw the devastating effect that collection of data can have in the 2016 Facebook-Cambridge Analytica scandal, when personal data from Facebook profiles were collected and used for political advertisement. This has resulted in data-protection laws and policies that have been implemented, like the General Data Protection Regulation (GDPR). However, the on-device security in the field of IoT is insufficient. The individuals want to protect their privacy, the company wants to protect its monetary assets and society wants to protect its infrastructure. This can be achieved by data protection and restricting access to devices.

Greater interconnectedness between devices, between systems and in infrastructure at large leads to more efficient cities. The general goal is to build a world where more can be done in less time, using fewer resources. These so-called "smart cities", characterized by the considerable use of linked devices, depend upon adequate cybersecurity.

With smart cities and the use of IoT-products, new sustainable solutions for everyday tasks can be found. As an example, consider an IoT washing machine. With possibilities to monitor electricity generation and consumption, the washing machine can decide to run its program with the intent to use sustainable power. This would result in less demand on the power generated from fossil fuels, which in turn leads to a more climate-friendly cleaning. Another example would be directing traffic in a town by the use of autonomous vehicles and IoT traffic lights. With improved flow of traffic and reduced traffic jams, carbon dioxide emissions would be cut down significantly. In conclusion, IoT-products could contribute to a more sustainable society against the climate crisis.

Hacking and Evaluating the Cybersecurity of an Internet Connected 3D Printer

Linus Backlund and Linnéa Ridderström

***Abstract*—Over the last few years, internet-connectivity has come to be an expected feature of professional 3D printers. Connectivity does however come at a cost concerning the security of the device. This project aimed to evaluate the cybersecurity of the Ultimaker S5 3D printer. The system was tested for the most likely and severe vulnerabilities based upon a threat model made on the product. The results show that the system's local webapplication is vulnerable to some common web-attacks that allow the attacker to perform actions on the victims printer.**

***Sammanfattning*—De senaste åren har internetuppkoppling blivit en självklar funktion hos professionella 3D skrivare. Uppkoppling kommer dock ofta på bekostnad av enhetens säkerhet. Detta projekt syftade till att utvärdera cybersäkerheten hos 3D skrivaren Ultimaker S5. En hotmodell gjordes och systemet penetrationstestades baserat på denna. Resultaten visar att enhetens lokala webbapplikationen är sårbar för några vanliga webattacker som låter attackeraren exekvera oönskade funktioner på offrets skrivare.**

***Index Terms*—Cybersecurity, Offensive Security, Penetration Testing, Hacking, Internet of Things, 3D printer.**

Supervisor: Pontus Johnson

TRITA number: TRITA-EECS-EX-2021:181

I. Introduction

This project aimed to evaluate whether the connected 3D printer Ultimaker S5 is a secure product. This was decided by penetration testing the device itself, its local webapplication and its communication with software interfaces. Due to legal reasons the cloud service was only briefly tested by non-intrusive means.

II. Background

More things and devices in peoples' everyday life are being connected to the internet; becoming a part of the so called Internet of Things (IoT). The IoT is a network of interconnected devices, through the Internet, that with its rapid expansion is easing the lives of mankind; providing service at the press of a button. Companies are already beginning to implement this in manufacturing, for example with IoT 3D printers. But what if someone was to get their hands on a company's new secret product while it's still in prototyping? By having more IoT devices in manufacturing, the manufacturing process in itself will become more efficent and easy, but the matter of cybersecurity will grow in importance. One method to strengthen the cybersecurity is to literally do the opposite of preventive security work; to ethicly hack a device. By finding the holes in the security, they can be plugged before they become a problem. The work done here is intended to make our connected world just a little bit safer and provide a basis upon which others can perform security testing on similar devices.

A. Selection of System

The main reasons for choosing the Ultimaker S5 as the system to hack in this paper was due to the recent emergence of connected 3D printers and the lack of previous work on the category regarding ethical hacking. A compromised 3D printer could lead to theft of valuable data, sabotage on property or, in a worst case scenario, physical harm to individuals. Ultimaker is one of the leading companies in the industry, producing advanced, high end desktop 3D printers. Their connected printers can host both a local webserver as well as connect to a cloud service. The Ultimaker S5 printer is affordable enough to be sold in large numbers and has the connectivity features that are expected of future printers. It has large attack surfaces as it connects to browsers, computer software, a mobile application and a cloud service through its several Application Programming Interfaces (APIs). These were the reasons for choosing this system specifically.

B. Related Work

No previous published penetration tests on 3D printers could be found. Previous work on IoT devices have been performed by other students from the same institute as the authors. From these, methods used in evaluating smart garages [1], vacuum cleaners [2] and an OBD-II device [3] were adapted to this untested product category.

III. Theory

Some tools and concepts that may not be familiar to the reader are briefly introduced before moving on to the methodology of the project.

A. Tools

1) Nmap: Nmap is a reconnaissance tool that is used to retrieve information about a networks layout and the devices connected to it. Common uses are finding devices on a network and scanning them for open network ports [4].

2) Wireshark: Wireshark is a very popular tool used to monitor network traffic. It can both gather the data and browse data gathered by other software [5].

3) PCAPdroid: This android application can be used to gather network data on a mobile device that can later be loaded and viewed in Wireshark [6].

4) Burp Suite: This popular web security tool has a wide range of features but are mainly used in this project to run automated scans for web-vulnerabilities and to do manual web communication manipulation [7].

5) Pentest-Tools.com: This online tool runs automated scans for encryption vulnerabilities and bad implementations on webservers [8].

6) Binwalk: Binwalk is a software that can disassemble and unpack many types of firmware files [7].

7) Firmwalker: Once a firmware has been unpacked, the filesystem can be automatically searched for interesting files with Firmwalker. It looks for files containing hardcoded passwords, credentials, IP-addresses etcetera [7].

B. Concepts

1) HTTP and HTTPS: HyperText Transfer Protocol or HTTP is an application level internet protocol. It is used to send requests and receive responses. There are several *methods* that can be used and indicate what the request means. Examples of these are GET and POST where GET means to retrieve data and POST means to upload data. A request can contain a payload and *headers* which provide additional options for how the request is to be handled. HyperText Transfer Protocol Secure or HTTPS is an encrypted version of HTTP. It can use any version of the older Secure Socket Layer (SSL) or, more commonly today, the newer and more secure Transport Layer Security (TLS). SSL and TLS are cryptographic protocols. This means that they are designed to be used to encrypt, and therefore secure, data communication [9].

2) Uniform Resource Locator (URL): A URL is way to reach a specific service on the internet. A example of a URL is*http://www.example.com/index.html*. The *http* part means that the HyperText Transfer Protocol (HTTP) is used, *www.example.com* refers to the specific server on the internet to send the request to and *index.html* is the specific path or resource to be requested [9]. It is also possible to add additional request parameters following a *?*-character to pass data to the receiving request handler.

3) Cross Origin Resource Sharing (CORS): CORS is a policy implemented by webservers that instructs browsers when and when not to allow requests to be sent to them.. CORS can be said to resemble a whitelist of sorts. If say, domain *A.com* is on the servers CORS-list, then any request made by *A.com* may be sent to the server without any issue. If *A.com* was not on the list however, the browser would not allow the request to be sent in order to prevent falsified requests to be sent from malicious websites on the users behalf [10].

4) Representational State Transfer-API (REST-API): A REST-API is a set of rules for communicating with a web service. For example, rules are set on what sort of requests can be made, what data-format these should have and much more. The REST-API consists of several endpoints, which are combinations of an HTTP method and a URL-path, that corresponds to different responses and different actions [9]. For example sending a 3D-file with POST to the path */print_job* could upload a file to be 3D printed.

5) Extensible Markup Language (XML): XML is a flexible structured language designed to store and transport data. This is done by having data being placed inside start and end-tags, which indicate what the data is to be interpreted as. It is very flexible and the designers of a system that use XML-files can define their own tags to accommodate their needs [9].

6) Hypertext Markup Language (HTML): HTML is an XML-based language that, when loaded by a browser, is interpreted as a website [9].

7) iframe: The iframe is a tag used in HTML to load another website inside a frame. A webserver can tell the browser that its content is not allowed to be loaded within an iframe with the HTTP header *X-Frame-Options* [11].

8) Flask: Flask is a framework for the Python language for building webserver APIs in a structured way [12].

9) SWUpdate: SWUpdate is a framework for updating Linux-based firmware. The updates come as *.swu* files and can be signed, making it possible to confirm whether the source is reliable and whether it has been tampered with [13].

IV. Methodology

This section gives an account of, as well as an explanation of, the chosen approach and several overarching methods used in this project - in chronological order. However, the specifics of individual tests are disclosed in the later section, VII.

A. Information Gathering

Information Gathering can be done from two different perspectives and depending from which the selected system is to be examined from, different gathering methods can be used [7].

Black Box: This perspective refers to the tester not being given any information on the system other than what is publicly available. This can include, for example, documentation, firmware and network details [7].

White Box: In this perspective the tester has been given additional internal information such as an architecture overview or elevated privileges by the vendor. This requires an agreement with the vendor [7].

In this project the vendor did not respond to a request to perform a white box analysis and so the black box perspective was taken. It did however turn out that a lot of information that typically only is provided in white box testing could be retrieved from publicly available sources from the vendors and from examining the product itself.

The following are the methods and tools that were used in understanding and mapping the printers functionality.

1) Using the system: Using the system to be tested is the obvious first step. It is important to get to know the printer to later be able map out all functionalities and technologies used.

2) Public documentation: Publicly available documentation could be an invaluable source of information during black box testing. It might not only speed up the Threat Modeling but also reveal some of the systems internal workings.

In the case of the Ultimaker printer, the vendor has some interesting documentation and information on their website regarding their security measures and procedures [14], the network configuration of the printer [15], systems and protocols used [16], firmware recovery procedures [17] as well as the firmware files themselves [18]. These documents need not necessarily be correct or complete but are a very good starting point for investigating the system

On the local webserver hosted by the printer, documentation of the printers main API could be found. By analyzing traffic as described in section IV-A3 the endpoints for documentation on the additional APIs could be found too. This documentation contained all possible endpoints, methods and models for expected payloads.

3) Network and traffic analysis: For analyzing the printer from a network perspective, *Nmap* was used to discover the printers open ports. *Wireshark* was used to analyze the traffic between the users computer and the printer. It was also installed on the printer itself to monitor outgoing traffic to the remote servers. Using the mobile application *PCAPdroid*, traffic between the mobile application and printer could be exported and analyzed in *Wireshark*.

4) Automated scans: Manual testing is a must for many attack-vectors but automated scans can help in mapping out the system. *Burp Suite Professional* was used for web-scanning and *Pentest-Tools.com* [8] was used for traffic encryption analysis.

5) Firmware analysis: Analyzing firmware is a good way to get familiar with the system architecture. The printers developer mode was activated to allow log in over Secure Shell (SSH). The filesystem was then explored to understand how the software components work together. The firmware was also downloaded from Ultimakers website [18] and extracted using *Binwalk*. The *Firmwalker* tool was used to automatically scan for interesting files such as configuration files or clear text passwords.

6) Source code analysis: When having retrieved the firmware as decribed in section IV-A5 the source code was available for analysis. This is very useful to an ethical hacker as it is possible to understand how the often hidden backend processes inputs. One example is understanding how the system handles requests to the APIs.

B. Threat-Modeling

The Information Gathering process took place in conjunction with the Threat-Modeling process. This is because the collected data from the Information Gathering process is directly used in Threat-Modeling. Threat-Modeling is a process used to make a thorough security analysis of a system. The goal is to create a *Threat Model* of the system from which risks and flaws in a system, that may make it vulnerable to different types of threats and attacks, can be identified [19]. A Threat Model is the end result of this process and can be seen as a complete mapping of the system and the potential threats against it. Threat-Modeling is commonly used in the early design stage of development in order to identify where to implement security controls and countermeasures to prevent potential attacks to and exploitation of the system [7].

There are six steps to the Threat-Modeling. *Identifying the Assets*, *Creating an IoT Device Architecture*, *Decomposing the IoT Device*, *Identifying Threats*, *Documenting Threats*. and *Rating Threats* [7].

The first step, Identifying the Assets, is about identifying and documenting the subsystems in the system that can be attacked and exploited. When this is done, the process moves on to the second step of Creating an IoT Device Architecture. There are three sub-steps to the second step. Firstly all *User Cases* need to be identified, which is the documentation of how a legitimate user would use the system and its offered functionalities. Then the different types of *Technology* that the system in consideration utilizes are noted, and lastly a simplified diagram of the data flow is created [7].

In the third step to the process, Decomposing the IoT Device, potential intrusive entry points to the system are documented and the diagram from the previous step is expanded upon with more detailed descriptions of the data flow. After this is done, the accumulated data acquired in the previous steps can be used in the fourth step, Identifying Threats, to identify potential risks and threats that may, for example, pose a security, safety or privacy risk for the user, the network or the vendor. A common way of identifying as well as categorizing these threats is through the **STRIDE** model [7].

STRIDE: The STRIDE model is an acronym for six categories of threats, namely:

- **Spoofing identity:** To gain access by falsifying ones identity or by pretending to be someone else.
- **Tampering with data:** To modify data without authorization. For example, by manipulating network packets in transaction.
- **Repudiation:** The user's ability to deny a specific action they have performed.

- **Information disclosure:** To display private or unauthorized data.
- **Denial of Service:** Making the target system nonoperational, for example by overloading it with requests.
- **Elevation of privileges:** A user or attacker with limited privileges gains more privileges, for example those of an admin.

From these, potential threats may be identified through brainstorming [7].

In the last step of Rating the Threats, the identified threats are ranked in severity; meaning they are ranked on their likelihood of success as well as on their possible impact on the system. There exist different rating systems to rate the threats by and a common one is the **DREAD** rating system [7].

DREAD: DREAD is a mnemonic for the questions asked when assessing the threats and stands for:

- **Damage Potential:** If this was to be exploited, how bad would the damage be?
- **Reproducibility:** With what ease can the attack be reproduced?
- **Exploitability:** How easy is it to carry out the attack?
- **Affected Users:** How many people will be affected?
- **Discoverability:** How easy is it to discover this vulnerability?

Each question is ranked from 1-3; where 1 is considered low risk, 3 is considered high risk and 2 is a medium risk in between the two extremes. The total score can range from 5 to 15 where a total score of 5-7 is considered low risk, 8-11 is medium risk and 12-15 is high risk [7].

C. Threat Traceability Matrix

The Threat-Modeling process was followed by the creation of a Threat Traceability Matrix. A Threat Traceability Matrix is an interpretation of the Threat Model and lists the previously identified potential attacks to the system [19]. It is also a reader-friendly way to display the results of the attacks tested on the system. The aim of the Threat Traceability Matrix is to showcase to the reader that all relevant and important attacks have been considered when probing the security of the system. In the matrix, for each and every attack displayed, the following aspects should be disclosed [20].

- **The Attack:** The potential attack to be tested.
- **Attack Surface:** Where is this attack conducted against the system? For example, from the browser?
- **Affected Asset:** What asset is compromised?
- **Threat Agent:** Who would carry out this attack?
- **Attack Goal:** What is hoped to be achieved?
- **Probability of Success / Success Rate:** An estimation of how likely it is that the attack succeeds.
- **Attack Impact:** If an attack was to be successful, how severe are the consequences in that case?
- **Attempted?:** If the attack in question was attempted.
- **Test Results:** What was the results (if attempted)?

Threat Traceability Matrix should also contain references to other related work or sources to showcase the validity of an attack [20].

D. Penetration Testing

After the risks and potential threats to the system were identified and ranked through the Threat-Modeling process and after concrete ways to attack and exploit the system were formulated through the Threat Traceability Matrix, the project moved on to the Penetration Testing of the system. This is the phase where concrete tests, the attacks that have the most promise according to the Threat Traceability Matrix, are tested against the security of the system [21]. More on the penetration testing can be read in section VII.

E. Reporting and Responsible Disclosure

All found vulnerabilities have been responsibly disclosed to the manufacturer. All relevant information have been sent through the channel requested by Ultimaker [14]. They have also been provided with sufficient instructions and code for proof of concept to be able to quickly and easily reproduce the attacks themselves. The publication of found vulnerabilities is held until the threats have been mitigated or 90 days has passed, which is suggested to be more than enough according to the Dutch National Cyber Security Centre [22]. A meeting has been held between the authors and the vendor to discuss the findings.

V. The System under Consideration

The Ultimaker S5 running firmware version 6.3 is a connected desktop 3D printer marketed as a simple and integrated solution to prototyping in an office environment. On the software side the device hosts several REST-APIs that are used by the local webserver, the desktop print-preparation software ”Cura”, the mobile application ”Ultimaker” and by the cloud service. The web interface, ”Digital Factory”, allows the user to monitor the printer, control the print-queue and change a few settings. This can be reached by entering the printers IP-adress in a browser on the local network or by logging in through Ultimakers cloud service. Some of the functionality can also be accessed in Cura and the Ultimaker app over the local network or by logging in to the cloud service through Cura. This is however, if the included firewall has not been manually activated restricting the printer to cloud connectivity only.

Apart from the main API documented directly on the local webserver, there are four additional ones used in the background. The cluster-API manages most of the general printer functionality such as managing printer-groups, print queues and cloud connection. The fault-API is for error management. The material-station-API is for getting the status of the material manager peripheral if connected. The hermes-API is for the setting up and managing of subscriptions to printer notifications. These subscriptions are set up over the local network but the notifications are delivered anywhere over the cloud.

As far as hardware connections go, the printer has a USB-port for physically uploading printfiles and firmware, an NFC-reader for material-identification and a DIN-connector for additional Ultimaker peripherals such as an active air-filter and material-manager. On the front there is a touchscreen on which most settings are configured. The device can be connected to the network over Ethernet or wirelessly over WiFi [23].

A developer mode can be activated on the printer allowing full root access through an SSH connection on the local network. When this setting is activated it is displayed on the printers screen.

VI. Threat Model

By analyzing the system, a complete Threat Model was created through the Threat-Modeling process. The reader is referred to the Appendix A, B and C for the outcome to the *Identifying the Assets*, the two sub-steps *Technology* and *User Cases* in *Creating an IoT Device Architecture* and *Decomposing the IoT Device* steps respectively. To gain an understanding of how the systems are interconnected a simplified Data-flow diagram of the 3D printer was created, shown in figure 1.

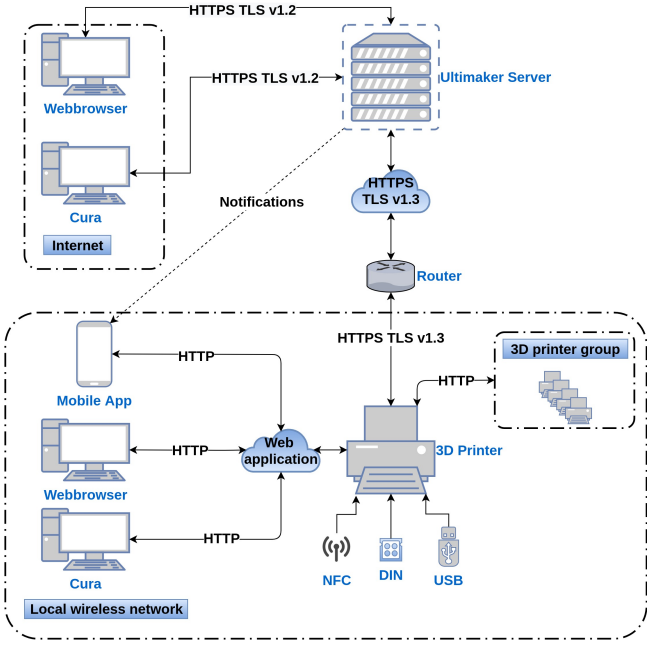


Fig. 1. A simple diagram showing how different applications are interconnected in the system and how data flows between them.

A. Delimitations

A security evaluation needs to be comprehensive in order to declare a system secure or insecure, but due to several factors the scope of this evaluation had to be limited. Therefore a set of delimitations were made. The delimitations were set on three separate occasions. Before and after the Threat-Modeling process as well as during the Penetration Testing of the system.

Delimitations that were set before the Threat-Modeling process included not practicing attacks that may be illegal. This meant not testing the cloud service and the server of the company that have produced the 3D printer as this would be illegal without permission. Though it was decided that a careful Cross Site Request Forgery (CSRF) attack could still be made on the Ultimaker account. This is because a CSRF attack would not be illegal if it was tested only against the attackers own user account.

Delimitations that were set after the Threat-Modeling process were due to the sheer size of the system under consideration in conjunction with the limited time frame of the project. It was decided that only the local network was to be explored. Add-ons like Ultimaker Air Manager and Ultimaker S5 Material Station were not to be exploited. This also included directly attacking the mobile app and the Cura program as this would be a whole project in itself.

During the penetration testing it was decided that potential threats and attacks that are not very serious in terms of impact or have low success rate are neglected. Time restraint played a role in this.

B. Identifying Threats

To identify potential threats to the system under consideration, the **STRIDE** model was used. The potential threats identified with the **STRIDE** model are listed below.

Spoofing

- Spoof different components in the system. Things to spoof may include the server or the 3D printer.
- Analyzing authentication and controls in the system, for example in the API, and learn how to bypass them. The goal is to identify ways to reach automatic trust relationships.

Tampering with data

- Change the packets that are sent through the local network.
- Delete, upload or start print files on the 3D printer.
- Altering the settings to enable more ways into the 3D printer. Potential tools would be Clickjacking, Cross Site Scripting (XSS) and CSRF.
- Forge requests.

Repudiation

- Through the use of SSH if there are no logs.
- Identify and utilize what practices can be done without logging abilities.

Information disclosure

- Intercept traffic on the local network sent from the web browser, Cura, the mobile app and leak that traffic.
- Malicious file upload through the SD card, USB- or Ethernet Port.

Denial of service

- Use the forgotten password functionality to lock out user from the Ultimaker account or through other means like changing the password (such as CSRF).
- Capture packets and discard them to block communication on the local network.

Elevation of Privileges

- An attacker connects to the 3D printer through the browser or by downloading their own mobile application.

- Brute force login on the user homepage account.

In conjunction with the STRIDE model, additional threats were identified by consulting the *OWASP top 10 Web Application Security Risks*. The OWASP top 10 Web Application Security Risks is a list, with a broad consensus, over the most common reasons and ways web applications are being exploited today and was created by the OWASP Foundation. Its purpose is to aid developers and others with web application security [24].

For this project, the following categories,

- **A1:2017-Injection:** The attacker tricks the application to carry out commands, code or queries by injecting them. This can include attacks like CSRF, Clickjacking, Code- and OS Command injection.
- **A3:2017-Sensitive Data Exposure:** The attacker utilize that data is poorly protected and encrypted.
- **A4:2017-XML External Entities (XXE):** Attackers tricks vulnerable XML processors with a hostile XML file.
- **A5:2017-Broken Access Control:** Restrictions are not properly enforced, for example in the API.
- **A6:2017-Security Misconfiguration:** Improper or incomplete configuration of security measures, for example, for frameworks.
- **A7:2017-Cross-Site Scripting XSS:** The attacker runs a hostile script due to no proper validation.
- **A8:2017-Insecure Deserialization:** The attacker sneaks in code in deserialized objects which can lead to code execution [24]. Serialization is the process of converting objects into a format that is more easily stored in databases, deserialization is the opposite process [25].

were of interest from the *OWASP top 10 Web Application Security Risks*.

C. Documenting Threats

In the tables I to V, the main threats that were identified are documented and expanded upon. These are the threats that were considered critical and within the scope of the delimitations. For threats that were delimited and therefore not continued with in this project, the reader is referred to see Appendix D.

TABLE I
THREAT 1

Threat Description	**The attacker could install malicious firmware or applications on the 3D printer.**
Threat target	3D printer firmware
Attack techniques	Plant malicious firmware by sideloading upon a firmware update. Or by uploading through the USB port, the SD card and/or the DIN connector. Or to the web application by using Command injection. In worse case, the attacker compromises the 3D printer by turning off the temperature barriers; posing a fire hazard.
Countermeasures	Validation. Whitelist commands.
STRIDE Threat	Tampering with data. Elevation of privileges.
OWASP Threat	A6:2017-Security Misconfiguration

TABLE II
THREAT 2

Threat Description	**The attacker uses common web exploitation attacks and social engineering to gain access to a user account or bypass authentication.**
Threat target	Ultimaker Account. The local web application.
Attack techniques	Uses attack methods like CSRF, XSS and Clickjacking to gain access to or lock out the user from their account. OS Command and Code injection is also a possible method to gain shell access or execute OS commands.
Countermeasures	Use anti-CSRF tokens and purification. Automatic log out after a set time. X-Frame-Options header.
STRIDE Threat	Tampering with data. Denial of service. Elevation of privileges. Spoofing.
OWASP Threat	A1:2017-Injection, A4:2017-XML External Entities (XXE), A7:2017-Cross-Site Scripting XSS.

TABLE III
THREAT 3

Threat Description	**The attacker could fake a 3D printer to achieve a goal, e.g. to enable other attacks.**
Threat target	Embedded web application. Mobile application.
Attack techniques	An attacker mimics an Ultimaker 3D printer.
Countermeasures	Certification.
STRIDE Threat	Spoofing. Tampering with data.
OWASP Threat	A5:2017-Broken Access Control.

TABLE IV
THREAT 4

Threat Description	**The attacker exploits social engineering and the local API to change settings, steal data or execute other type of attack on the printer.**
Threat target	Local web application and API
Attack techniques	Social engineering. XSS. CSRF. Features of the API.
Countermeasures	Authentication for use of API even on local networks. Anti-XSS/CSRF purifying of requests.
STRIDE Threat	Spoofing. Elevation of privileges. Tampering with data.
OWASP Threat	A5:2017-Broken Access Control, A7:2017-Cross-Site Scripting XSS,

TABLE V
THREAT 5

Threat Description	**The attacker manipulates settings that allow more access to the printer, e.g. to enable other attacks.**
Threat target	Embedded web application.
Attack techniques	Clickjacking. Corrupt settings-files or print files. Social engineering.
Countermeasures	X-Frame-Options header. Validation.
STRIDE Threat	Elevation of privileges.
OWASP Threat	A1:2017-Injection, A5:2017-Broken Access Control.

D. Rating Threats

To determine how damaging these threats would be in practice the threats were ranked in accordance with the **DREAD** rating system. The results from rating the threats in this project are displayed in Table VI.

TABLE VI
DREAD - IN THIS TABLE, THE RATING OF THREATS WITH DREAD IS DISPLAYED. THE FIRST LETTER IN EACH QUESTION IS USED TO SIGNIFY THE QUESTION ASSESSED.

	Threat 1	**Threat 2**	**Threat 3**	**Threat 4**	**Threat 5**
D	3	2	1	3	2
R	2	3	3	3	2
E	2	3	3	3	2
A	1	2	1	2	1
D	2	2	3	2	2
Total	10	12	13	10	9

E. Threat Traceability Matrix

The Threat Model was interpreted and a Threat Traceability Matrix with appropriate attacks was created, as can be seen in the table VII. Some attacks were discovered during the Penetration Testing of other attacks and added later. The table VII only contain the attacks that went on to the Penetration Testing. For other considered attacks that was later disregarded due to the set delimitations, one is refereed to the appendix D. Every attack in the Threat Traceability Matrix displayed in table VII, including the theory behind, then will be thoroughly explored in section VII.

Risk Matrix:
To better visualise the estimated gravity of the attacks, a risk matrix was created as can be seen in Table VIII. A risk matrix is a diagram that visualises the risk by considering two key factors, *Probability of Success* and the *Severity* of the attack. [21]

TABLE VIII
RISK MATRIX - IN THIS TABLE, THE ATTACKS ARE RATED. SEV STANDS FOR "SEVERITY" (OF THE ATTACK), WHILE PR STAND FOR "PROBABILITY OF SUCCESS" (OF THE ATTACK).

PR / SEV	**Negligible**	**Marginal**	**Critical**	**Catastrophic**
Certain				
Likely			Clickjacking	CSRF
Possible		Billion Laughs, Quadratic Blowup	Code Injection, XSS, XXE	OS command Injection
Unlikely		POODLE, Heartbleed	Firmware Modding	
Rare				
Eliminated				

VII. PENETRATION TESTS

Following are the penetration tests performed on the printer. Under each attack, a brief description of it, the method of testing, the results and their implications in short are described. Further analysis of the results and what they mean for the project at large can be found in sections VIII and IX.

A. Cross Site Request Forgery (CSRF) on local webserver

Cross Site Request Forgery is an attack where an attacker tries to forge requests to a site on the victims behalf. The attack could be delivered as a link that either directly sends the request or loads a malicious webpage from which the request is performed. Possible outcomes of a CSRF attack could for example be making the victim unknowingly perform a bank transfer or deleting a user-account. Common CSRF vulnerabilities are crafted links using URL-parameters to perform certain actions or exploiting misconfigured CORS policy [32].

Method: To find CSRF vulnerabilities the *Burp Suite* automated webscanner was run at all levels of intrusion on Digital Factory on the local webserver. The site was manually searched for URL-parameters. The printers APIs were explored and manually tested for CORS misconfigurations. Traffic was monitored through the inspection tool in the browser. To understand backend functionality the source code for relevant APIs and webpages was read.

Result: The Burp Suite scanner indicated nothing of interest. The local webserver did not use any URL-parameters. The main API was fully documented on the local webserver and the location of the documentation for the other APIs could be found by analyzing traffic and testing URL-paths. This information, containing not only all endpoints for all APIs but also the model for each payload allowed for manually sending requests.

The main API required authorized credentials. These could be requested but required a confirmation through the physical touchscreen on the printer. The other four APIs required no authorization at all which made them interesting surfaces for further testing.

A website was written that made HTTP-requests through the Javascript fetch method. The requests were however prevented in the browser due to the header *Access-Control-Allow-Origin* not being present in the response. The APIs do not have a CORS implementation resulting in the browser preventing all cross origin requests to them by default.

To try and bypass this restriction the *no-cors* mode was used in the request. No-cors indicates that the browser can send the request, without preflight i.e. agreeing with the server that the request is to be sent and only with *"safe methods"* and *"safe headers"*. Using no-cors limits the requests to *GET*, *HEAD* and *POST*-methods and the response from the server can not be accessed by the javascript code [10]. In testing using no-cors it was discovered that the local webserver had no controls whatsoever of the received requests and subsequently processed them as if they were legitimate. In other words, a vulnerability had been discovered and could be exploited for a CSRF attack.

TABLE VII
THREAT TRACEABILITY MATRIX - IN THIS TABLE, THE THEORETICAL PART OF THE THREAT TRACEABILITY MATRIX IS DISPLAYED, BUT ONLY OF THE ATTACKS THAT WENT ON TO THE PENETRATION TESTING.

No #	Attack	Attack Surface	Asset	Threat Agent	Attack Goal	Success Rate	Attack Impact
1	CSRF	Browser	Browser, Ultimaker Account & Local web-application.	Unauthorized External Attacker	To execute unwanted actions and functions such as changing the password, email or printing a file [7].	High	High
2	Clickjacking	Browser	Browser, Ultimaker Account & Local web-application.	Unauthorized External Attacker	To socially engineer the user into performing an action they are not aware of, for example, by making them click a button on a webpage [26].	High	Medium
3	XSS	Browser	Local web-application.	Unauthorized External Attacker	To gain session cookies for a session hijacking, other types of sensitive information like key logs or make the printer inoperational to the user [7].	Medium	Medium
4	Command injection	Browser	Firmware, 3D printer.	Unauthorized External Attacker	To execute OS commands that can give the attacker admin privileges that the producer didn't originally intend them to have or access to passwords [7].	Medium	High
5	Code injection	Browser	Firmware, 3D printer.	Unauthorized External Attacker	The attacker's code makes the target execute an action that is not a part of its base programming [27].	Medium	High
6	XXE	Browser, USB	Local web-application.	Unauthorized External Attacker	To upload hostile XML files to exploit vulnerable code or dependencies [24].	Medium	High
7	POODLE	Internet Communication	Remote server communication.	Unauthorized External Attacker	Force the printer and cloud to communicate over insecure HTTPS so that it can be listened to or manipulated [28].	Medium	High
8	Heartbleed	Internet Communication	Remote server communication.	Authorized External Attacker	Steal data using the OpenSSL heartbleed vulnerability [29].	Low	Medium
9	Billion Laughs	Browser, USB	Local web-application.	Authorized External Attacker	This is a Denial of Service attack meant for breaking the service by overloading the memory [30].	Low	Medium
10	Quadratic Blowup	Browser, USB	Local web-application.	Authorized External Attacker	This is a Denial of Service attack meant for breaking the service by overloading the memory [30].	Medium	Medium
11	Firmware Modding	Internet Communication, USB	Firmware	Authorized External Attacker	Uploading malicious firmware through unrestricted file upload [31].	Low	High

Some endpoints require a Universally Unique Identifier (UUID) version 4 associated with a specific machine or printjob to be known. These are essentially impossible for an external attacker to guess as there are $5.3 \cdot 10^{36}$ possibilities and so are not exploitable by this attack alone.

***Discussion**:* The response could not be read and the accessible endpoints were limited to the *GET* and *POST* methods. None of the GET-requests performed any action other than returning values but some POST-requests could be used to make the printer do various tasks.

The malicious webpage could be hosted on a remote server. It did however have to be served over insecure HTTP to avoid the browser blocking it due to *Mixed Content*. Most browsers warn users that the website is insecurely hosted but by the time a victim could notice the subtle note, the code would already have been executed.

Two example websites were written as proof of concept and can be found in appendix E. The first one uploads a file to the printers queue through the cluster-API as depicted in figure 2. If it was first in line and the correct material was loaded, it would be automatically started.

The second was a little more complicated and is depicted in figure 3. Utilizing *PCAPdroid* and *Wireshark*, the process for setting up a subscription for notifications on the mobile app was analyzed. A Javascript API, also found in the appendix, based on the express framework was written intended to mimic a printer. This was done by giving acceptable responses to the endpoints requested by the application. By manually entering the hosts IP-address and the port on which the API was served in the mobile app a subscription request with its unique token could be captured. This was then integrated in a website as the previous example. By this method a subscription was possible to set up on the victims printer providing the attacker with notifications, sent over the cloud, about every print started or stopped on the machine. If the setting for image sharing was not turned off, the attacker would also receive preview images of the models, an example of which is depicted in figure 4.

Parts of the API endpoints for setting up a cloud connection can be accessed with the exploit. This could be explored as future work to potentially get even more access to the system.

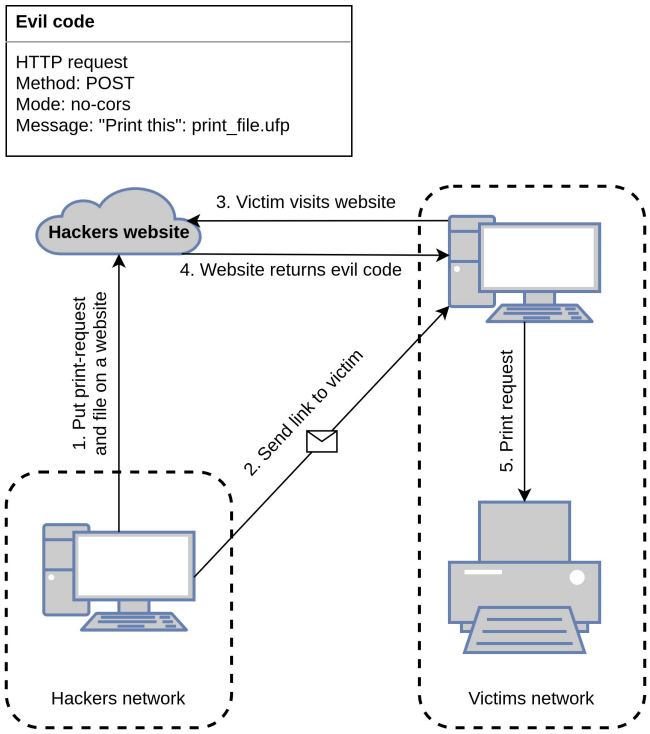


Fig. 2. Diagram of how the CSRF vulnerability can be used to upload a printjob to the printer

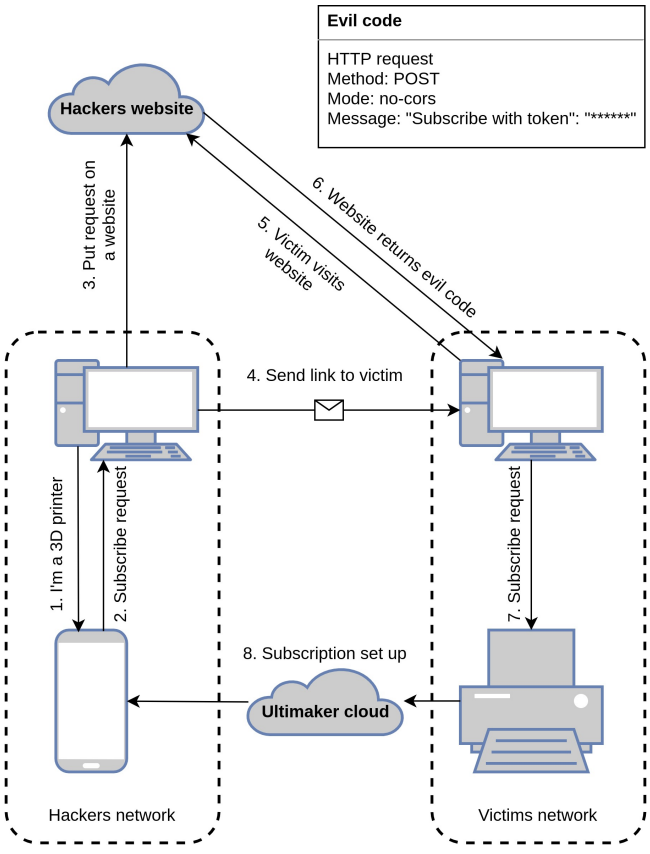


Fig. 3. Diagram of how the CSRF vulnerability can be used to set up a notification subscription from the printer to the mobile application

Fig. 4. Example of a preview image sent in a notification to the mobile application

B. Cross Site Request Forgery (CSRF) on Ultimaker cloud

This test is essentially the same as the previously mentioned one on the local webserver but instead focused on Ultimakers cloud service. This includes the Digital Factory interface as well as the users login page.

Method: Just as in testing the local webserver in VII-A the automated scanners provided in *Burp Suite* were run at passive and light active levels against the pages. They were searched for URL-parameters and the CORS configuration was examined.

Result: The automated scans did not return much of interest and none of the pages used URL-parameters to make any changes. CORS appeared to be correctly implemented on most sites. Only one had its *Access-Control-Allow-Origin* header set to *. After testing it by manually sending GET and POST requests it was clear however that it used some other means of authentication and that a successful bypass would not only affect the user but also the entire service infrastructure. Due to delimitations based on legal reasons this was not further investigated.

A webpage was written to attempt to make no-cors requests just as in the attack on the local webserver. Measures had been taken to prevent the no-cors exploit as a POST-request with no-cors mode changing a users email returned a *415: Unsupported Media Type* error.

The pages for printer settings required a printer groups unique id to be known and so were not tested.

Discussion: The Ultimaker cloud service is not vulnerable to any obvious CSRF attacks. Since the backend is closed source there is no way of ensuring it does not contain obscure bugs making it possible to forge requests without a lot more testing. The reason for the previously discovered no-cors vulnerability on the local webserver not working on the cloud is because no-cors only allows for specific *Content-Type* headers. This means that this header can not be set to the value required by the server and so it returns an error. There could possibly be even more checks implemented if this was in some way bypassed. Further work could be exploring these measures using other software to manually craft the requests.

C. Clickjacking

Clickjacking is a method in which the attacker constructs a website that loads another target website inside an *iframe* tag, makes it completely transparent and carefully positions it over some decoy content. The effect of this is that the victims believe themselves to be visiting one page and clicking a button but actually performs an action on the targeted, invisible webpage positioned on top [26].

Method: *Burp Suites* automated scans were used to search for frameable pages and its built in tool *Clickbandit* was used to do manual tests.

Result: The automated scan reported a lot of exposed pages on the local webserver. *Clickbandit* was used to manually test all clickable actions on the cloud based Digital Factory as well as the user pages. This resulted in the attacker being logged out. At closer inspection the cloud service used the *X-Frame-Options* header set to *sameorigin* preventing most clickjacking attacks. On the local webserver however the content was loaded into the iframe without issue. And with that, a vulnerability had been found.

Discussion: As a proof of concept a website exploiting Clickjacking on the local webinterface was developed which can be found in the appendix at F. Using this exploit an attacker could trick the victim into toggling the share image setting that could be used in synergy with the subscription attack described in VII-A. This example is depicted in figure 5 where the target website been made slightly opaque for demonstration purposes.

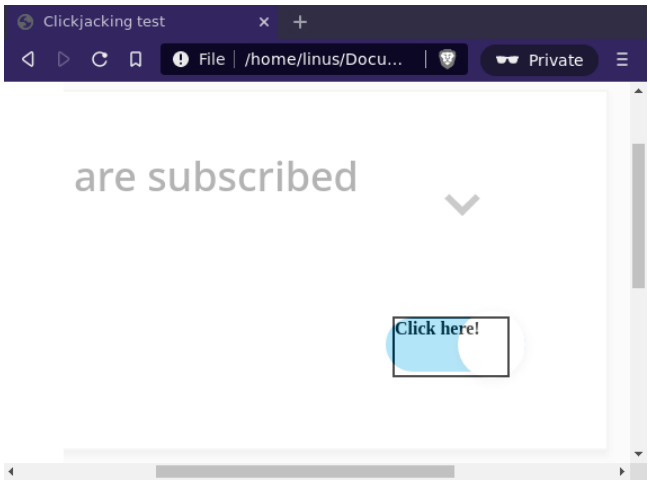


Fig. 5. Example of the clickjacking attack used to manipulate a user to toggle a setting. The usually transparent target page has been made slightly opaque for demonstration purposes.

Using a more complicated two click approach an attacker could potentially make a user abort, duplicate or move a printjob in the queue.

D. Cross Site Scripting (XSS) on local webserver

Cross Site Scripting is an attack where a user inputs data on a server that is then reflected back to the same or another user. If this payload is carefully crafted it could be interpreted as valid code that is loaded and executed in the page on the victims browser [24].

Method: The *Burp Suite* scanners were run on the local webserver and manual XSS was attempted in the form of specifically crafted printjob files and material profiles that were then uploaded to the printer.

Result: The *Burp Suite* scans returned a few low severity threats. The webserver used Javascript dependencies, *jQuery 1.11.3*, *jQuery 1.8.0* and *handlebars 4.0.5*, that have known vulnerabilities. They are integrated in the wifi setup page and the API documentation pages which are unlikely to be viewed and therefore used in an attack.

The manual tests resulted in what looked in the source code of the rendered webpages to be valid code but it was not interpreted as javascript.

Discussion: No XSS vulnerabilities were found and the file information displayed on Digital Factory appears to be purified. It is not likely that the local webserver is vulnerable to XSS.

E. OS Command injection

Command injection is an attack applicable to a situation where a user input is processed directly in a system environment. This means that operating system commands are interpreted as such and executed on the target machine. A command injection vulnerability could potentially give the attacker a lot of access to the system [33].

Method: All of *Burp Suites* automated scans were run on the local webserver and some user inputs were time delay tested and analyzed to discover if the payloads were executed on the system. The source code was read to understand how the system processes the inputs.

Result: The scan showed no results of interest. Manual tests on the input field for renaming the printer would not allow certain characters. Upon inspection of the page a Javascript validation function was found. This filter was bypassed by capturing a request, modifying and replaying it which resulted in it being sent. Upon inspecting the source code it was revealed that all handling of the input was done within a Python script without any functions making system calls. The Flask framework used also verified that the input was a string.

Discussion: No case of command injection could be found either by intrusive automated scans or manual testing. The system is very unlikely to be vulnerable to command injection.

F. Code injection

Code injection is similar to command injection in that it is based on missing validation of input. It does ultimately, in contrast with command injection, execute code within the parsing application rather than system commands [27].

Method: The backend python source code was examined for how the inputs were processed. Whether they were put inside common functions for code execution such as *eval()* and *exec()* or if they were validated to be the desired types.

Result: The input box for renaming printers was examined in the source code. It was validated in the backend Flask application against a Flask model requiring it to be a String. The backend code for a majority of the API endpoints was examined and this strict validation was consistent throughout

the entire codebase. No *eval()*, *exec()* or other vulnerable functions were found.

Discussion: The backend code appears to be robust regarding input validation to avoid code injection. It is not likely that any input has been missed but they would have to be tested and investigated one by one to be entirely sure.

G. XML External Entity (XXE)

XML External Entity is an attack where an XML document is crafted to load external resources. This could either be used to retrieve local files from the system that parses the file if the XML is reflected to the user or used to trick the system into loading content from a remote server [24].

Method: To test this attack, common XML payloads requesting a Python *SimpleHTTPServer* local webserver hosted on a computer were injected into the XML material profiles and the XML-files in the .ufp printfile. Ten materials and 17 printfiles were produced with the payload located in different elements in the files. They were uploaded to the printer while monitoring incoming requests to the server. Additionaly the source code was read to identify how the files were parsed.

Result: None of the crafted XXE-injected files yielded any positive result whatsoever. In reading the source code it was discovered that the python module *ElementTree* was used without the optional *ElementInclude*. This means, according to the Python blog [30], that the parser is not vulnerable to XXE but potentially to internal entity expansion-based attacks such as Billion Laughs and Quadratic Blowup. More on this in VII-H.

Discussion: The printer is not vulnerable to XXE attacks because the parser does not expand external entities. This is good design by the vendor and the only way to perform this attack would be to first find a vulnerability in the *ElementTree* module itself which is unlikely.

H. XML Internal Expansion attacks (Billion Laughs & Quadratic Blowup)

As discovered in VII-G the device might be vulnerable to internal entity expansion attacks such as Billion Laughs and Quadratic Blowup. These attacks are both based on the idea of making the device expand an entity in an XML file until it runs out of memory and the service breaks.

Method: To test for this vulnerability a typical Billion Laughs payload was created by following the previously mentioned Python blogpost [30]. It was injected into three different entities in three different XML materialfiles. The same was done for Quadratic Blowup. These were then uploaded to the printer and the webinterface was monitored.

Result: Two of the files for each attack were successfully accepted as valid XML and uploaded to the printers materials storage. The webservice went down for approximately one minute returning a *502: Bad Gateway* error. New files were created that expanded even more but this did not increase the downtime.

Discussion: It appears the device is vulnerable to these attacks but in simply uploading the files no more harm than one minute of downtime could be done. It could be possible that if a print is started, requesting a malicious material, another effect could be achieved. To do this however, a material profile claiming to be a newer version of an existing material that the printer is likely to be loaded with would have to be created. This comes with a few challenges, the greatest being that the official profiles comes with signature files. This signature functionality would therefore need to be investigated and broken. This could be interesting for continued testing.

I. Firmware modding

Firmware updates are common in IoT products. This is because updating them is easy due to them being connected, and because the products need to be continually compatible with the rest of their ecosystem as it grows. They are also a great way to implement security patches but an update can be a vulnerability in itself. Firmware modding could give the attacker full control of the product provided they find a way to deliver the modified firmware.

Method: Ultimakers documentation will be read as well as the source code handling the firmware update. The update packages will be examined.

Result: The updates come as *.swu* files. The device use the *SWUpdate* framework. All downloaded firmware updates are signed and verified against Ultimakers servers. Files provided over USB or SD card are not verified.

Discussion: The use of a framwork such as *SWUpdate* that signs the firmware is a secure method. To bypass this the *SWUpdate* framework would have to be broken. Not verifying the physically delivered firmware is not very secure. The SD slot is inaccessible without removing the bottom plate of the device. The USB port might however be exploited by someone with temporary physical access but no network access. On the other hand, the attacker could then also change the network configuration and activate developer mode in the settings to get access to the system instead.

J. POODLE & Heartbleed

POODLE and the Heartbleed attack are both attacks focused on transport layer encryption such as SSL and TLS. POODLE is an attack where the attacker tries to inject packets into the setup of an encryption channel. The goal is to downgrade the encryption used to a version with known vulnerabilities to be able to perform a man in the middle attack. Heartbleed is a vulnerability in the OpenSSL implementation of TLS that leaks memory from the targeted system [29].

Method: The remote server that the printer connects to was identified using *Wireshark* and tested for the vulnerabilities by the online *Pentest-Tools.com* vulnerability scanner in light mode. *Wireshark* was also used to analyze *Client hello* messages in HTTPS connection setup to determine the TLS versions supported by the printer.

Result: Two servers were identified. One connected to the *api.ultimaker.com* domain and the other to a *Google cloud storage*. The servers were scanned and discovered not to be vulnerable to Heartbleed. The oldest version of encryption protocol supported by the Ultimaker server was TLSv1.2 and by the Google storage, TLSv1.0. A captured Client hello message showed that the printer only supports TLSv1.2 and TLSv1.3. In other words, the printer is not vulnerable to POODLE or Heartbleed. The results also showed that the Ultimaker server was not vulnerable to an additional eleven encryption-related attacks tested by the tool.

Discussion: None of the vulnerabilities were found on the Ultimaker server and downgrade attacks like POODLE are prevented by the printer in the case of the Google storage server.

VIII. Results

The summarized results to the penetration testing of the system can be found in table IX, the continuation of the Traceability Matrix.

TABLE IX
THREAT TRACEABILITY MATRIX - IN THIS TABLE, THE RESULTS ARE SUMMARIZED AND THE THREAT TRACEABILITY MATRIX THEREFORE IS COMPLETED.

No #	Attempted?	Summarized Result
1	Yes	A CSRF attack could be executed successfully on the local webserver. The vulnerability is that the webserver processes requests without validating them. CORS can be bypassed for POST-requests using no-cors mode.
2	Yes	A clickjacking attack could not be performed on the Ultimaker cloudinterface but was possible to execute successfully on the local webserver. The vulnerability is that no *X-Frame-Options* header was used instructing the browser not to allow the attack.
3	Yes	No XSS vulnerability could be found. It is unlikely that the system is vulnerable to XSS.
4	Yes	In examining the system it was concluded that it is not vulnerable to Command Injection attacks.
5	Yes	In examining the system it was concluded that it is not vulnerable to Code Injection attacks.
6	Yes	In examining the system it was concluded that it is not vulnerable to XXE attacks.
7	Yes	In examining the systems it was concluded that it is not vulnerable to the POODLE attack.
8	Yes	In examining the systems it was concluded that it is not vulnerable to the Heartbleed attack.
9	Yes	Billion Laughs attacks could be executed successfully on the system. The impact was however no more than one minute of downtime. It is possible that other means of executing the payload can yield more severe implications.
10	Yes	Quadratic Blowup attacks could be executed successfully on the system. The impact was however no more than one minute of downtime. It is possible that other means of executing the payload can yield more severe implications.
11	Yes	In examining the system it was concluded that it is not vulnerable to firmware modding when downloading updates. Physically delivered firmware, however, is vulnerable but was deemed a less severe scenario as there are simpler ways to control the system given physical access.

IX. Discussion

While the product with its default settings is not very secure, Ultimaker provides great advice on their website on how to make it so [14]. It appears the printers current functionality is by design but the local webserver is lacking in security. The firewall setting on the printer eliminates all the found vulnerabilities and is highly recommended by Ultimaker. We therefore recommend, however, that it should be activated by default and prompt the user about the risk when being disabled. This would still allow for those cases that require the communication to stay on the local network without exposing the average user to security threats. It is entirely possible that the less technically experienced user does not dare touch the advanced settings or does not want to as it seemingly disables features.

The mobile application has not been updated since 2018 and the way it sets up subscriptions need to be remade. As of now it relies entirely on a unique token to connect a printer to the app. The more secure way to do this would be binding the printer to a user on the Ultimaker cloud service. This has already been implemented in the cloud version of Digital Factory and could be incorporated into the mobile app.

All code found in the printers firmware was well written, well structured and generally followed secure practices. It is clear that the developers are competent and so it has to be assumed that they simply did not realize what risk an insecure webserver could mean.

Future work could be further testing for XML internal expansion attacks as the attacks were successful but no way of deployment with severe consequences could be found. If given allowance by the vendor, the cloud service would be valuable to test further, for example for CSRF. During the penetrationtesting of the attacks OS Command Injection and Code Injection, a potential vulnerability grounded in Insecure Deserialization were found. Namely that, in the firmware, the vulnerable Python package Pickle was used at one point to serialize objects. This potential threat and attack was documented in the delimited part of the Threat Traceability Matrix and can be seen in more detail under Appendix D, but due to the extensive code and time limitation this potential vulnerability was not explored. Therefore future work could be exploring this more extensively. It could also be of some interest to further look for XSS vulnerabilities, manually test for CSRF on Ultimaker cloud and test other ways to deploy the XML Internal Expansion attacks.

A. The Sustainability and Ethics

No intrusive attacks or scans were performed on Ultimaker property in accordance with the delimitations and the law. The found vulnerabilities have been responsibly disclosed to Ultimaker as described in section IV-E. As Ultimaker is a leading company in the desktop 3D printing industry this public disclosure of the findings should not prove a threat to the industry but rather a support for companies within it to

design secure connected products. As described in section II-A the product was chosen because testing it for vulnerabilities would provide the most value to the future of connected printers as mentioned in section II. These findings could of course also be applied to other product categories in IoT.

X. Conclusion

The results of this project answers the researched question, whether or not the product is secure from a cybersecurity perspective. The Ultimaker S5 is a well designed product in many aspects but the local webserver is not secure. To a user with little knowledge about cybersecurity it is easy to not configure the 3D printer in a secure way and expose it to attacks. As Ultimaker already has secure alternative solutions so these problems are on their way to be fixed.

Appendix A
Identified Assets

Appendix B
Architecture Overview

Appendix C
Decomposing the IoT Device

Appendix D
Delimited Threats and Attacks

Appendix E
Code for CSRF proof of concept

Appendix F
Code for clickjacking proof of concept

Acknowledgment

The authors would like to thank their supervisor, Pontus Johnson, for providing valuable guidance and consultancy.

References

[1] M. Berner, "Where's my car? ethical hacking of a smart garage," Master's thesis, KTH, EECS, Stockholm, 2020.
[2] A. Larsson Forsberg and T. Olsson, "Iot offensive security penetration testing : Hacking a smart robot vacuum cleaner," Bachelor's thesis, KTH, EECS, Stockholm, 2019.
[3] H. Lindström and G. Marstorp, "Security testing of an obd-ii connected iot device," Bachelor's thesis, KTH, EECS, Stockholm, 2018.
[4] nmap.org. (2021, May) Nmap: the network mapper. [Online]. Available: https://nmap.org/
[5] The Wireshark Foundation. (2021, May) About wireshark. [Online]. Available: https://www.wireshark.org/
[6] Emanuele Faranda. (2021, May) Pcapdroid user guide. [Online]. Available: https://emanuele-f.github.io/PCAPdroid/quick_start.html
[7] A. Guzman and A. Gupta, *IoT Penetration Testing*. Birmingham, UK: Packt Publishing Ltd, 2017.
[8] Pentest-Tools.com. (2021, Apr.) Powerful penetration testing tools, easy to use. Pentest-Tools.com, Bucharest, Romania. [Online]. Available: https://pentest-tools.com/home
[9] Larry Peterson and Bruce Davie, *Computer Networks: A Systems Approach*, 6th ed. Amsterdam, NL: Elsevier, 2019, ch. 1, 7, 8, 9. [Online]. Available: https://book.systemsapproach.org/index.html
[10] Evert's Dugout. (2020, mar) Common no-cors misconceptions. [Online]. Available: https://evertpot.com/no-cors/
[11] Mozilla and individual contributors. (2021, May) <iframe>: The inline frame element. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTML/Element/iframe
[12] The Pallets organization. (2021, May) Flask. [Online]. Available: https://flask.palletsprojects.com/en/2.0.x/
[13] Stefano Babic. (2021, May) Swupdate framework. [Online]. Available: http://swupdate.org/
[14] Ultimaker BV. (2021, Apr.) Ultimaker security. Ultimaker BV, Utrecht, The Netherlands. [Online]. Available: https://support.ultimaker.com/hc/en-us/articles/360020928359-Ultimaker-Security
[15] ——. (2021, Apr.) Networking ports and domains used by ultimaker. Ultimaker BV, Utrecht, The Netherlands. [Online]. Available: https://support.ultimaker.com/hc/en-us/articles/360012113719-Networking-ports-and-domains-used-by-Ultimaker
[16] ——. (2021, Mar.) Faq: Network and security protocols utilized by ultimaker printers and digital factory. Ultimaker BV, Utrecht, The Netherlands. [Online]. Available: https://support.ultimaker.com/hc/en-us/articles/360018732359-FAQ-Network-and-security-protocols-utilized-by-Ultimaker-printers-and-Digital-Factory
[17] ——. (2021, Jan.) Firmware recovery procedure for the ultimaker s3 and the ultimaker s5. Ultimaker BV, Utrecht, The Netherlands. [Online]. Available: https://support.ultimaker.com/hc/en-us/articles/360019088780
[18] ——. (2021, Apr.) Update the ultimaker s5 firmware. Ultimaker BV, Utrecht, The Netherlands. [Online]. Available: https://support.ultimaker.com/hc/en-us/articles/360011545559-Update-the-Ultimaker-S5-firmware
[19] Synopsys Editorial Team. (2016, Jul.) The 5 pillars of a successful threat model. Synopsys, Mountain View, California. [Online]. Available: https://www.synopsys.com/blogs/software-security/5-pillars-successful-threat-model/
[20] Pontus Johnson. (2020, Apr.) Method. NSE Lab, Stockholm, Sweden. [Online]. Available: https://nse.digital/pages/thesis_guidelines/method.html
[21] ——. (2020, Apr.) Threat traceability matrix. NSE Lab, Stockholm, Sweden. [Online]. Available: https://nse.digital/pages/thesis_guidelines/threat_traceability_matrix.html#threat-traceability-matrix
[22] National Cyber Security Centre. (2018, Oct.) Coordinated vulnerability disclosure: the guideline. Nationaal Cyber Security Centre (NCSC), The Hague, The Netherlands. [Online]. Available: https://english.ncsc.nl/binaries/ncsc-en/documents/publications/2019/juni/01/coordinated-vulnerability-disclosure-the-guideline/WEB_Brochure-NCSC_EN.pdf
[23] 3DVerkstan. (2021, Apr.) Ultimaker s5. 3DVerkstan AB, Solna. [Online]. Available: https://3dverkstan.se/ultimaker-s5-2/
[24] OWASP® Foundation. (2017) Owasp top ten web application security risks. OWASP® Foundation, Bel Air, California. [Online]. Available: https://owasp.org/www-project-top-ten/
[25] Vickie Li. (2021, Nov.) Hacking python applications. Pentest-Tools.com, Bucharest, Romania. [Online]. Available: https://medium.com/swlh/hacking-python-applications-5d4cd541b3f1
[26] Gustav Rydstedt. (2021, Apr) Clickjacking. OWASP® Foundation, Bel Air, California. [Online]. Available: https://owasp.org/www-community/attacks/Clickjacking
[27] Weilin Zhong, Rezos. (2021, Apr) Code injection. OWASP® Foundation, Bel Air, California. [Online]. Available: https://owasp.org/www-community/attacks/Code_Injection
[28] Tomasz Andrzej Nidecki. (202, jun) What is the poodle attack? [Online]. Available: https://www.acunetix.com/blog/web-security-zone/what-is-poodle-attack/
[29] OWASP® Foundation. (2021, Apr) Heartbleed bug. OWASP® Foundation, Bel Air, California. [Online]. Available: https://owasp.org/www-community/vulnerabilities/Heartbleed_Bug
[30] Brian Curtin. (2013, Feb.) Threat traceability matrix. [Online]. Available: https://blog.python.org/2013/02/announcing-defusedxml-fixes-for-xml.html
[31] OWASP® Foundation. (2021, Apr) Unrestricted file upload. OWASP® Foundation, Bel Air, California. [Online]. Available: https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload
[32] ——. (2021, Apr) Cross-site request forgery prevention cheat sheet. OWASP® Foundation, Bel Air, California. [Online]. Available: https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html
[33] Weilin Zhong. (2021, Apr) Command injection. OWASP® Foundation, Bel Air, California. [Online]. Available: https://owasp.org/www-community/attacks/Command_Injection

IoT Security Assessment of a Home Security Camera

Ida Kols and Nina Hjärne

***Abstract*—The amount of IoT devices in society is increasing. With this increase there is an inherently higher risk of hackers exploiting the vulnerabilities of such a device, accessing sensitive personal information. The objective of this project was to assess the security level of a home security camera through finding vulnerabilities and exploiting them. The method used for this was to analyze the system and its communication, threat model the system to identify threats, perform vulnerability analysis and exploit the vulnerabilities through penetration testing. The attacks on the system did not succeed and the system was declared secure in the vulnerability analysis. From the aspects tested in this project, it can be assumed that safety precautions have been taken to secure the home security camera from malicious hackers.**

***Sammanfattning*—Antalet IoT-produkter i samhället ökar och med fler och fler uppkopplade produkter i våra hem ökar risken att hackare utnyttjar produkters sårbarheter för onda avsikter, till exempel för att komma åt känslig personlig data. Målet med det här projektet var att hitta sårbarheter i en säkerhetskamera för hemmet, attackera dem och utifrån resultatet bedöma hur säker produkten är. Detta gjordes genom att analysera systemet och dess kommunikation, göra en hotmodell för att identifiera hot, genomföra sårbarhetsanalys och sedan penetrationstesta hoten. Hackningsattackerna misslyckades och produkten bedömdes som säker i sårbarhetsanalysen. Utifrån de aspekter som testades i projektet kunde det bedömas att grundläggande säkerhetsåtgärder vidtagits för att skydda säkerhetskameran från hackare.**

***Index Terms*—security camera, cyber security, ethical hacking, vulnerabilities, threats, penetration testing, IoT, privacy**

Supervisor: Pontus Johnson

TRITA number: TRITA-EECS-EX-2021:182

I. Introduction

The digital revolution has changed the way we live. The progress of Internet of Things (IoT) is remarkable and today we are surrounded by connected devices that simplify our everyday life. In addition to mobile phones; nowadays, cars, traffic lights, security cameras, washer machines, etc, are often internet-connected.

However, IoT devices have brought many security threats into our homes. Connected devices can collect a great amount of data through sensors that hackers might get access to. Therefore, it is important that the developers put a lot of effort into security issues connected to the device. Unfortunately, this is often not as prioritized as needed [1].

It is important that no malicious hacker can take advantage of the hidden vulnerabilities in the devices. Ethical hackers can counter this by finding the vulnerabilities and report them to the developers and warn users about the risks.

A. Objective

The objective of this project was to assess the security level of an IoT device of choice. The selected device is the SpotCam Sense home security camera [2].

B. Methodology

The methodology is based on a structure described by Georgia Weidman. Weidman divides penetration testing into the steps pre-engagement, information gathering, threat modeling, vulnerability analysis, exploitation, post-exploitation and reporting [3]. The pre-engagement phase was not included in this project because of the decision not to contact the SpotCam company and instead treat the device as a black box. All information about the product was discovered in the information-gathering phase. This included searching for information on the internet, port scanning and studying the communication flow through the system. By threat modeling, the system's possible security threats were defined and ranked with a risk score. Threat modeling is further described in section III-E and the results are presented in section IV. The system was also analyzed for possible vulnerabilities that could lead to threats. In the exploitation phase penetration tests were performed on the vulnerabilities. Post-exploitation is when successful exploits lead to further testing. Reporting phase is when results are documented and any found vulnerabilities are reported to the developers. The tests performed on the system are described in sections V, VI, VII and VIII, and the results are presented in section IX.

C. Selection of system to explore

During the selection of the system to explore, some main areas were considered. How big of an impact a successful attack on the system could have, how likely it is that the developers have managed to secure the system and how big the possibilities of finding vulnerabilities in the system are. The last one depends on, for instance, the size of the attack surface.

The consequences of the SpotCam Sense being hacked could be a severe invasion of privacy or result in a break-in as the camera is meant to function as home security. The camera uses wireless communication over Wi-Fi and offers many different features. The camera can be accessed from both the mobile application SpotCam and a browser. This indicates a possibility for a large attack surface and high complexity of the product [2].

D. Delimitations

The early delimitations made were to exclude the attack surfaces of the hardware and the iOS mobile application. Due to the lack of physical attributes of the camera, the hardware attack surface was considered less important to test for vulnerabilities. The decision to exclude the iOS mobile application was based on the perception that Apple has made a bigger effort in constraining information than Android developers and that it therefore would be more time-efficient to focus on the Android version.

One important aspect to keep in mind when deciding which attacks to implement is the law. It is illegal to hack anything that you are not the owner of, which means that attempts to penetration test, for example a cloud server, could not be made [4].

II. Background

A. Spotcam Sense

The studied IoT device is the home security camera SpotCam Sense. Cloud stored camera feed is available to the user for 24 h through the related mobile applications, both iOS and Android, and web application. The storage time limit can be increased to a maximum of 30 days with a subscription. The camera connects to the local network through Wi-Fi. A microphone, speaker and sensors for humidity, temperature and luminosity are some additional features of the product. The camera has the ability to change between two modes, client mode and access point (AP) mode. AP mode is used during the setup and installation of the camera and can be motivated by a switch on the camera [2].

B. Testing environment

Kali Linux that is a Linux distribution specifically designed for ethical hackers was used as testing environment on the computer. The operating system has several penetration testing tools pre-installed.

C. Software tools

1) Wireshark: A network protocol analyzer that comes pre-installed in Kali Linux. It can expose detailed information about the network traffic and the data packets [5].

2) aircrack-ng: A program used to assess Wi-Fi security and is a collection of several tools including packet sniffers. It can be used for monitoring, attacking, cracking and testing [6].

3) Ettercap: A tool for man-in-the-middle attacks and can be used for ARP poisoning and network analysis, among other things. Comes pre-installed in Kali Linux [7].

4) mitmproxy: A man-in-the-middle proxy that can examine HTTP and HTTPS traffic [8].

5) Nmap: Network mapper (Nmap) is a tool used in network discovery and can perform different scan types such as UDP scan or TCP scan, service detection or operating system detection. By the use of raw IP, Nmap can determine many characteristics of a network or a host [9].

6) Nessus: An automated vulnerability scan that scans the host for known vulnerabilities [10].

7) Hydra: A network login cracker that can be used for several protocols. The user can include passwords and usernames lists as input and hydra will try every login combination. Hydra is included in Kali Linux [11].

8) Enjarify: A tool that translates Dalvik byte code to Java code [12].

9) JD-GUI: A decompiler that displays code of Java classes. Used for analyzing methods [13].

10) MobSF: Mobile Security Framework (MobSF) is a Python tool for automated static analysis and security assessment of application binaries [14].

D. Previous Testing

Not much previous testing has been done on SpotCam Sense but a couple of reports has been made. In 2016, Pen Test Partners published a test of SpotCam that presented a few vulnerabilities. The Nmap scan showed three open TCP ports, Telnet (23), HTTP (80) and RTSP (554). The tester could get access to the web directory which showed the password to the Wi-Fi connected to SpotCam. The source code of the website contained a link to a snapshot of the camera feed to be accessible with that link. The test also reported vulnerabilities in the web server configuration and well-known Telnet passwords [15].

In 2017, another testing was posted on the-gadgeteer.com which proved a security development of SpotCam Sense Pro. The port scans generated a whole different set of open ports, with the Telnet port as the only remaining one from the Pen Test Partners test. The most important conclusion drawn from this test was that the camera did not seem to run a local webserver anymore [16].

III. Theory

A. Internet of Things

The Internet of Things (IoT) is the category of devices that can communicate and be controlled over an internet connection. IoT devices contain CPU, memory, firmware and a network interface which makes interfering with other connected devices possible. The system resembles a computer system and the software is often more complex than the purpose of the device requires. Because of this, vulnerabilities are almost inevitable [17].

B. Protocols

Systems communicate over the network through a set of different languages called protocols. Each protocol has its own way of initiating a connection, encrypting data, formatting data, detecting errors and correcting them, and terminating the connection. The different protocols that have been involved in this project are IP, ARP, TCP, UDP, Telnet and HTTP/HTTPS [18].

1) IP: Internet Protocol (IP) is the most common protocol for communications between systems on different networks. All devices connected to a network have an IP address where the first part represents the network it is connected to and the second part represents the devices' location on the network. IP has currently two versions, IP version 4 (IPv4) and IP version 6 (IPv6) [18].

2) ARP: Address Resolution Protocol (ARP) is used when resolving a Memory Access Control (MAC) address connected to a specific device on the local network. MAC addresses are required to transfer information from one system to another, as it represents the receiver. An ARP request is sent to the receiver's IP address and its MAC address is sent back with an ARP response. The MAC address is then saved in a Content Addressable Memory (CAM) table together with addresses to other systems it communicates with. This way an ARP request is not needed every time information is sent [18].

3) TCP: The Transmission Control Protocol (TCP) is a transportation protocol that establishes a reliable connection between the transmitter and the receiver. Before sending data the source connects to the destination and when the data is sent the connection is closed. This way it is ensured that the data will get to the receiver [18].

4) UDP: User Datagram Protocol (UDP) is another transportation protocol but unlike TCP it is focused on communication with speed rather than reliability. UDP is often called a connectionless protocol because no formal connection between the host is made as with TCP. Instead, UDP often uses other methods to make the connection reliable [18].

5) Telnet: Telnet is an unencrypted application protocol and is foremost used for text-based communication and login. Telnet is often used via the terminal in order to connect to a terminal process on a remote host and control the system. Telnet has TCP 23 as its default port [19].

6) HTTP/HTTPS: The Hypertext Transfer Protocol (HTTP) is the most used protocol when browsing web pages. The protocol is based on requests and responses. The user agent sends a request to the server, wanting to get access to the server's content. The server determines what data the user can get access to by analyzing information sent by the host. HTTP has TCP 80 as its default port [20]. It is common to encrypt HTTP communication with Transport Layer Security (TLS), this is called HTTP over TLS (HTTPS) [21]. With an initial TLS handshake, authentication of both parts and negotiation of the encryption algorithm is made to establish a secure connection before data transfer [22]. The default port for HTTPS is TCP 443 [21].

C. SaaS

SaaS stands for *Software as a service* and is a cloud-based software that supplies the software over the internet without the user having to download or install it on their local computer [23].

D. Firmware

The software that controls the device's applications and functions through hardware is called firmware. It consists of a bootloader, kernel, root filesystem and flash contents [24].

E. Threat modeling

Threat modeling is an exercise to identify attack surfaces and potential threats to a system in order to secure it. A common approach for threat modeling is the Microsoft threat modeling process consisting of six steps described in [24].

1) **Identifying Assets:** Enumerate all of the system's assets to identify the most probable areas for vulnerabilities.
2) **Device Architecture Overview:** Creating an overview of the system's components and functionalities to discover potential flaws in the design. Different Use cases can be created in order to do so.
3) **Decomposing Device:** Analyze the device's data flows and the technologies' trust boundaries to locate entry points into the system.
4) **Identifying Threats:** Identify and categorize threats using the STRIDE model described later in this section.
5) **Documenting Threats:** Document the description, attack techniques and any countermeasures for the identified threats from the STRIDE model.
6) **Rating the Threats:** By using the DREAD model and assigning the threats risk scores based on the DREAD aspects.

1) STRIDE: The threats a device may encounter can be divided into six categories: spoofing Identity, tampering with data, repudiation, information disclosure, denial of service and elevation of privileges. This is from the acronym called the STRIDE model and can be useful when determining possible threats. Spoofing is tricking someone/something that you are something that you are not, for example by using someone else's login credentials or by giving your computer another IP address. Tampering with data is to maliciously change data, for example, make a change to the communication data between devices. Repudiation is to hide actions that have been performed, for example, perform an attack that can not be traced. Information disclosure is when getting access to information that should be closed, for example, files in the system and private data. Denial of service (DoS) is to block a part of a system, for example, make a web server deny requests from a device. Elevation of privileges is when getting access to the system and get privileged rights; this could be used to make changes and destruction to the system [25].

2) DREAD: The DREAD model is a rating system that can be helpful when deciding which threats to explore further. The threats are ranked by rating the aspects: Damage potential, reproducibility, exploitability, affected users and discoverability. Damage potential rating indicates the severity of the damage, reproducibility indicates how difficult it is to repeat the attack, exploitability represents the complexity of the attack, affected users represents the number of people that are affected and discoverability is how hard it is to find the vulnerability. Every aspect is rated from 1-3 where 3 is high and the total risk score is the summation of all the aspects [26].

IV. Threat model

A. Identifying Assets

The systems identified assets are listed in Table I.

TABLE I
ASSETS

ID	Asset	Description
1	Security camera	This security camera consists of a camera lens, a microphone and a speaker. The device supports network protocols such as IP TCP, TLSv1.2 and v1.3, HTTPS, UDP, DNS, NTP. The camera has a switch which enables the camera to switch from Access Point (AP), used during installation, to client mode.
2	Camera	The camera lens provides live stream to the cloud service and can detect motion.
3	Microphone	Detects sound from the device surroundings.
4	Speaker	Can play voice messages from the user trough the web or mobile application.
5	Firmware	Based on Linux kernel. The systems feature settings are controlled via the firmware.
6	Vendor web application	Users can view live and cloud-stored camera footage from the vendors cloud SaaS platform. Requires username and password to log in. User can also configure settings, turn on sound and or motion detection notifications, use the 2-way communication, launch firmware updates and view additional information about the camera's surroundings like temperature and humidity.
7	Mobile application	Android and iOS applications are available. Connected to the vendor's cloud environment. Users can view live and cloud-stored camera footage from the vendor's cloud SaaS platform. Requires username and password to log in. Users can also configure settings, turn on sound and or motion detection notifications, use the 2-way communication, launch firmware updates and view additional information about the camera's surroundings like temperature and humidity.
8	Networking	Internet connections are done through Wi-Fi over the IEEE 802.11 protocol.

B. Device Architecture Overview

To help to create the overview of the device's architecture and communication flows, different use cases describing possible user actions were created.

The data flow diagram of the system sprung from the created use cases below is presented in figure 1.

Use case 1: User installs and sets up the camera to watch the camera feed.

1) User activates the device.
2) User creates a user on the website.
3) User downloads the installation file on a computer from the website.
4) User changes the device's mode to AP.
5) User connects the computer to the device's access point through a Wi-Fi-connection.
6) User connects the device to the wanted network by selecting it on computer.
7) User changes the device's mode to client.
8) User selects the camera on the account on the website and watches the camera feed.

Use case 2: User sends a voice message to the camera via the mobile application from a remote location.

1) User downloads the mobile application.
2) User logs into an account.
3) User records a message in the application.
4) The message is played on the camera.

Use case 3: User generates an update of the device's firmware at the local network.

1) User logs into an account on the website on a computer.
2) User selects the firmware update.
3) The device updates.

C. Decomposing Device

The identified entry points into the system are listed in Table II.

TABLE II
ENTRY POINTS

No.	Entry point	Description
1	Camera	Can be controlled via a browser or the mobile app. The existence of open ports on the device.
2	Vendor web application	The device communicates with the vendor's cloud SaaS platform over HTTPS, TLSv1.2 and TLSv1.3. Camera feed is stored on the cloud and is accessible from the mobile application and browser.
3	Mobile application	Both an iOS and an Android application are available. Requires username and password to log in for viewing the camera feed and executing other actions. Connects to the vendor's cloud environment.
4	Firmware	Controls the device. Listens on port 23 for telnet.
5	Wireless communication	Internet connections are made through Wi-Fi over the IEEE 802.11 protocol.

D. Identifying and Documenting Threats

The documented threats from the STRIDE analysis, their attack techniques and their countermeasures are presented in Table III.

E. Rating the Threats

The threats' risk scores from the DREAD model are shown in Table IV.

V. MAN-IN-THE-MIDDLE ATTACK

The concept of man-in-the-middle is that the attacker can capture the traffic between a client and a server by placing itself in the middle of the communication. Different methods for MITM attacks were conducted during the project, using an android mobile phone, two computers and the software tools Ettercap, Wireshark, aircrack-ng and mitmproxy [28].

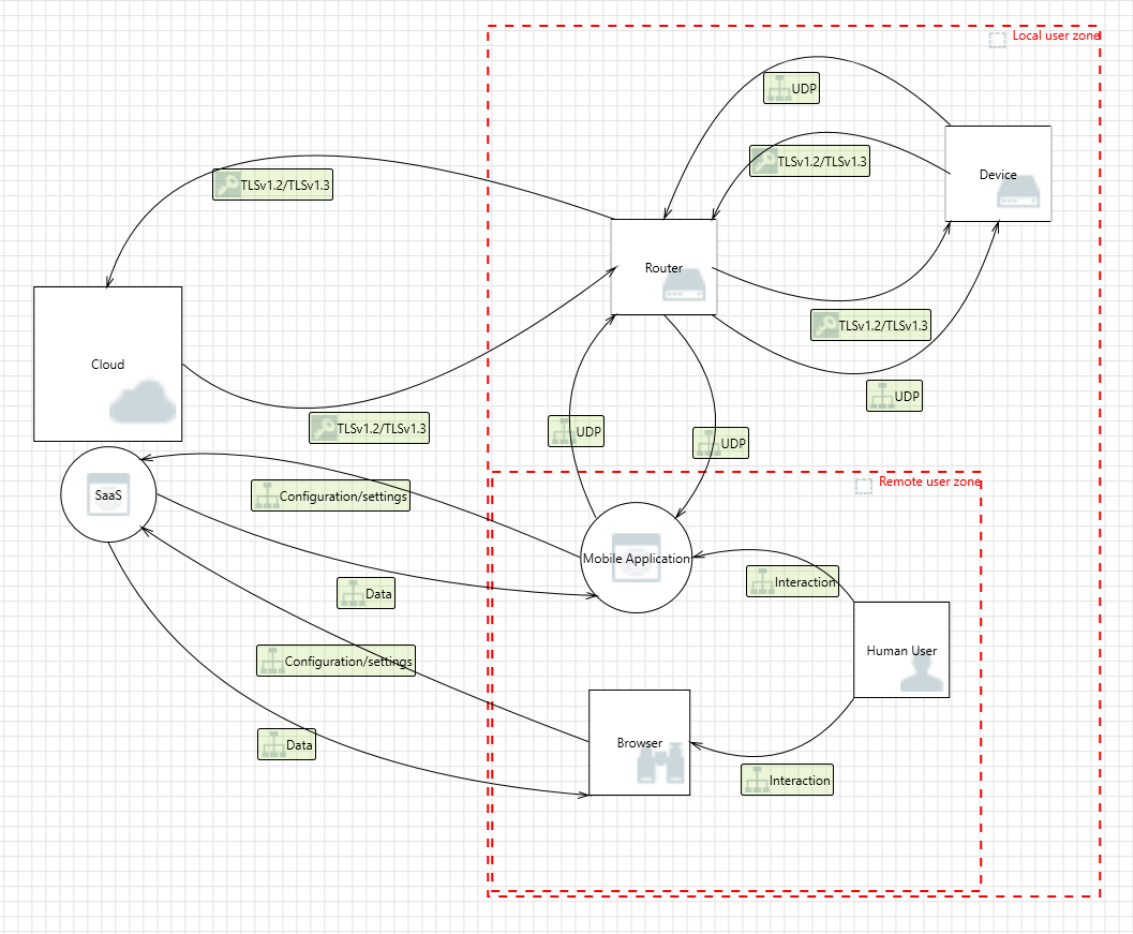


Fig. 1. Data flow diagram and architectural overview of the system.

A. Capturing traffic with Wireshark and network card in monitor mode

In order to map out the device's communication and data flow, this method was used to capture all the traffic and data packets sent to and from the SpotCam device. This method included setting one out of two computers' network card in monitor mode which means that the computer can monitor all the data packets sent to any device in a wireless network. Monitor mode was set by turning off the computer's network manager before using the tool aircrack-ng (command airmon-ng) to set monitor mode.

Wireshark was used to display all packets passing through the network. The second computer, the user, initialized the installation of the camera, then visited the SpotCam website[1] and executed different actions for the camera.

B. Using mitmproxy to route traffic through attacker's computer

This method was used for exploring how the mobile application behaves. The Android phone was configured to proxy, route all traffic through, the attacker's computer in the network settings. The tool mitmproxy was running on the computer and displayed the mobile phone's requests.

[1]www.myspotcam.com

C. ARP poisoning the victim and gateway with Ettercap

In an attempt to capture firmware during an update the method of ARP poisoning was used. ARP poisoning is performed by sending fake or spoofed ARP replies to two hosts communicating with each other in order to trick them both into associating the attacker's MAC address to the other host's IP address, hence putting itself in the middle of the communication [28]. This was done with the tool Ettercap where the SpotCam was victim 1 and the router was victim 2.

The traffic was then displayed in Wireshark. This method is in [24], presented along with the tool sslstrip for redirecting encrypted HTTPS traffic as HTTP traffic in order to obtain the firmware binary file during the update [29]. Unfortunately, sslstrip is no longer available for download and this attack had to be performed without it.

VI. Port scanning

Port scanning is done by trying to connect to the ports on a device to find which ports are open and which services they run since open ports could be potential entry points to the system. The tool used to perform port scanning in this project was Nmap. The first scan that was accomplished was a network scan to discover which hosts were up on the network and to identify their IP addresses. When the host's IP address

TABLE III
THREATS

Threat description	Threat type	Attack Techniques	Countermeasures
Attacker could eavesdrop on the device's wireless communication	Spoofing/Information disclosure	Performing a MITM-attack	Communication over protocols with encryption like HTTPS
Attacker could view the camera feed from the device	Spoofing/Information disclosure	Performing a MITM-attack and get access to credentials or session tokens to bypass authentication. Use session hijacking.	Multifactor authentication, encrypted communication [24]
Attacker could expose sensitive information in the application binaries (APK)	Information disclosure	Decompile the APK into Java code and analyze it to find information like username and password	Keep the APK from being accessible, avoid hard-coded sensitive information in the APK
Attacker could get hold of firmware during an update	Information disclosure	ARP poisoning and performing a MITM-attack to capture the data	Communication over protocol with encryption
Attacker could gain root access via open Telnet port	Elevation of privilege	Crack the password by a brute-force attack or testing for default usernames and passwords	Avoid using default or simple passwords and usernames for access
Attacker could cause a congestion attack to knock out the port service	Denial of service	Send a large amount of data to open port	Filtering input and validating the transmitter, make the device capable of handling a great amount of traffic [27]
Attacker could set up a fake access point to connect to the camera during installation	Spoofing	Creating a fake access point and social engineer the user into picking the attackers AP during installation of the device instead of the correct one	Ensure that the SSID requires an encryption key upon connection
Attacker could install malicious firmware on the device	Tampering	Try to interfere with firmware updates	Communication over protocols with encryption,
Attacker could take over the system and perform actions on the camera	Spoofing/Tampering/ Elevation of privilege	Performing a MITM-attack and get access to credentials or session tokens to bypass authentication and send requests	Multifactor authorization, lockout policy for limited log in attempts [27]

was known it was possible to perform a port scan on the host [30].

First, a scan with the flag `-sV` was run; a version detection scan to determine the versions of the services run by the open ports. The second scan was an operating system detection scan initiated with the flag `-O`. After enumerating all the open ports on the camera, the ports were singly examined using the flag `-A`; both version detection and operating system detection scan. The flag `-p-` was also used in all of the above-mentioned scans in order to scan all 65 535 ports. Otherwise, Nmap will perform the scan only on the 1000 most used ports [31]. The above scans are TCP scans but a UDP scan was done as well using the flag `-sU`. In addition to the port scanning, attempts were made to connect to the ports via a browser in order to detect a local web server on the device.

Since the camera can switch between AP mode and client mode, it was necessary to port scan the device in both modes. A simple port service scan with the camera in AP mode was conducted to identify open ports.

A. Vulnerability scans

Nmap also offers prepared scripts which can be used to conduct a vulnerability scan, with the flag `--script`, to identify vulnerabilities of a system. One such script named *vulners* was used to find potential vulnerabilities [32].

Nessus is a more commonly used vulnerability scanner with potentially more accuracy than the scripts in Nmap since the script risk missing some important vulnerabilities. Therefore, a Nessus scan was performed as well [28].

VII. PASSWORD CRACKING

Password cracking can be done by several methods. In this project, the method dictionary attack was used. In a dictionary attack usernames and passwords from a list are matched in every possible combination and tested on the system [28].

A. Dictionary attack on Telnet port with Hydra

In the port scanning process, an open Telnet port was found that required username and password for getting access to it. A dictionary attack was performed in order to find the right login credentials.

The program that was used for this purpose was Hydra. The following code was executed in the Linux terminal with lists of common usernames and passwords and the IP address of the device.

```
$ hydra -l <username list> -p \
<password list> <IP-address> telnet
```

The hydra test was run with four different lists. One list was created with common usernames and passwords and words with a connection to the device that came to mind, the list can be found in Appendix A. This list was used both as username and password. Two lists of common combinations were found on github with passwords connected to Telnet [33] and to

TABLE IV
DREAD

#	Threat	D	R	E	A	D	Risk score
1	Attacker could eavesdrop on the device wireless communication	2	3	2	1	3	11
2	Attacker could view camera feed from the device	2	3	1	1	3	10
3	Attacker could expose sensitive information in application binaries (APK)	2	3	3	3	3	14
4	Attacker could get hold of firmware during an update	2	1	2	3	2	10
5	Attacker could gain root access via open Telnet port	3	3	3	1	3	13
6	Attacker could cause a congestion attack to knock out the port service	3	2	3	1	2	11
7	Attacker could set up a fake access point to connect to the camera during installation	3	1	2	1	2	9
8	Attacker could install malicious firmware on the device	3	1	1	1	1	7
9	Attacker could take over the system and perform actions on the camera	3	3	2	1	2	11

BusyBox [34], these were tested as well. The last list was found in a blog post recommended for testing Telnet ports with service version BusyBox [35].

VIII. DECOMPILING THE MOBILE APPLICATION APK FILE

To analyze the mobile application in depth the application binaries were downloaded from the APK downloader Evozi[2]. APK, or android packages, are the binaries for android applications. To make the code analyzable these codes were translated to Java bytecode that then could be viewed with the Java decompiler JD-GUI. This was used to get an overview of the classes and to search for interesting hard-coded information [24].

Further analysis was performed by using the automatic vulnerability scan Mobile Security Framework (MobSF). This was done by starting a static analysis of the APK code.

IX. RESULT

A. Man-in-the-middle attack

1) Capturing traffic with Wireshark and network card in monitor mode: This test resulted in an overview of the device communication. From the results, it could be determined that the camera communicates with the vendor web application with TLSv1.2 (HTTPS) by using a temporary open port to connect to the cloud service. Figure 2 shows an image of the Wireshark output. These results were also used in the making of the device architecture overview presented in Figure 1.

2) Using mitmproxy to route traffic through attacker's computer: The results from mitmproxy showed only encrypted requests, HTTPS flows from the Android phone.

3) ARP poisoning the victim and gateway with Ettercap: Due to the lack of the previously mentioned tool sslstrip, the firmware binary file was not successfully acquired. The traffic that was captured in Wireshark includes the firmware that was sent from the cloud server over HTTPS but, this time, with TLSv1.3 unlike the previously captured communication between the camera and cloud server which was over TLSv1.2.

B. Port scanning

The output from the version detection scan in Figure 3 shows four open TCP ports on the camera. The Telnet port (23) service version was detected and identified as *BusyBox telnetd*. The 46850 port with unknown service can be interpreted as a dynamic port that connects to the cloud platform. This can be supported by the results from Wireshark in Figure 2 as well due to the display of the communication between this port and the cloud server [3].

An open Telnet port could make the system vulnerable to attacks by allowing an attacker to get root access to the system via the port. The possibilities for this have been tested and the results are presented in section IX-C.

The OS scan determined the operating system of the camera to be *Linux 2.6 - 3.5* as seen in Figure 4. It also discovered an additional open TCP port, 3000 which runs the service point-to-point (ppp). No new information about the open ports was detected when performing the scans on single ports.

The attempts to connect to a local web server on the device were unsuccessful, thus the lack of one could be assumed which the results from Nessus in section IX-B-1 also affirms.

The results from the UDP scan are presented in Figure 5. These results are from a UDP scan that was done much later in the process since the first UDP scan did not generate any open UDP ports.

The scan performed with the camera in AP mode generated the output in Figure 6. One additional port (5501) was discovered. Unfortunately, any further attempts to redo or proceed to scan on AP mode were unsuccessful since Nmap failed to detect the host as up again.

1) Vulnerability scans: The vulnerability scan using the vulners script in Nmap did not find any vulnerabilities in the system. The Nessus scan generated information about one vulnerability associated with the open Telnet port (23). The vulnerability is presented by Nessus accordingly:

"Using Telnet over an unencrypted channel is not recommended as logins, passwords, and commands are transferred in cleartext. This allows a remote, man-in-the-middle attacker to eavesdrop on a Telnet session to obtain credentials or other sensitive information and to modify traffic exchanged between a client and server."

[2]https://apps.evozi.com/apk-downloader/

[3]The port number is not the same because of the change of port before every new connection

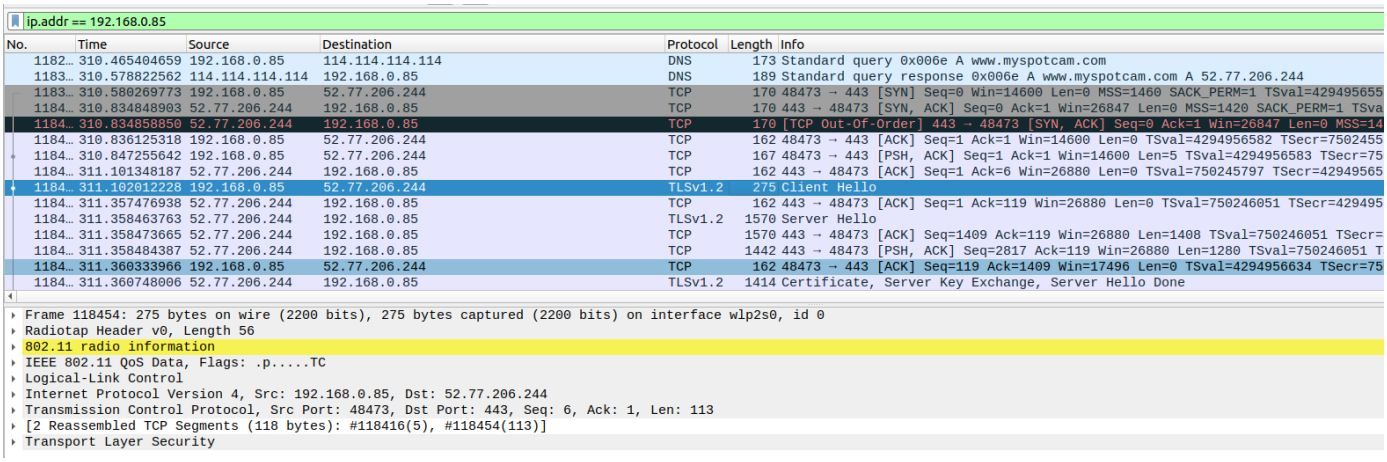


Fig. 2. Captured traffic in Wireshark shows the camera's communication.

```
# nmap -sV -p- 192.168.0.85
PORT      STATE SERVICE         VERSION
23/tcp    open  telnet          BusyBox telnetd
5503/tcp  open  fcp-srvr-inst2?
5552/tcp  open  unknown
46850/tcp open  unknown
MAC Address: 14:6B:9C:99:C9:59 (Shenzhen Bilian Electronicltd)
Service Info: Host: UlifeCam
```

Fig. 3. Nmap output from version detection scan.

```
# nmap -O -p- 192.168.0.85
PORT      STATE SERVICE
23/tcp    open  telnet
3000/tcp  open  ppp
5503/tcp  open  fcp-srvr-inst2
5552/tcp  open  unknown
MAC Address: 14:6B:9C:99:C9:59 (Shenzhen Bilian Electronicltd)
Device type: general purpose
Running: Linux 2.6.X|3.X
OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3
OS details: Linux 2.6.32 - 3.5
Network Distance: 1 hop
```

Fig. 4. Nmap output from operating system detection scan.

C. Password cracking

1) Dictionary attack on Telnet port with Hydra: Hydra did not find any combination of username and password from any of the lists that were tested that established access to the system. The test result to one of the dictionary attacks can be viewed in Figure 7.

D. Decompiling the mobile application's APK file

The analysis of the Java code in JD-GUI did not result in any findings of interesting data. MobSF gave the application a security score of 100 out of 100 and marks it as low risk. In the details, MobSF tells us that some information might be sent from the application in clear text for example with HTTP.

X. Discussion

The decisions made based on the threat model were to explore the highest-ranked threats according to the DREAD

```
# nmap -sU 192.168.0.85
PORT      STATE         SERVICE
1007/udp  open|filtered unknown
5351/udp  open|filtered nat-pmp
17946/udp open|filtered unknown
20817/udp open|filtered unknown
20851/udp open|filtered unknown
32774/udp open|filtered sometimes-rpc12
34358/udp open|filtered unknown
51905/udp open|filtered unknown
MAC Address: 14:6B:9C:99:C9:59 (Shenzhen Bilian Electronicltd)
```

Fig. 5. Nmap output from UDP scan.

```
# nmap -p- 192.168.234.1
PORT      STATE SERVICE
23/tcp    open  telnet
3000/tcp  open  ppp
5501/tcp  open  fcp-addr-srvr2
5503/tcp  open  fcp-srvr-inst2
5552/tcp  open  unknown
MAC Address: 14:6B:9C:99:C9:59 (Shenzhen Bilian Electronicltd)
```

Fig. 6. Nmap output from port scan on camera in AP mode.

model. It was discovered during the port scanning that a Telnet port 23 was open on the camera. This resulted in threat number 5 in Table IV; one of the high-risk threats. The results from the dictionary attack on the Telnet port indicate that no common default or easily guessed username and password is used for root access. No exploitation of the Telnet port was able to take place thus the camera can be considered to have strong security against this threat.

In the analysis of the APK, no hard-coded information was discovered and it was marked as secure by MobSF. These two threats were given the two highest risk scores and were therefore prioritized during the penetration testing. During the man-in-the-middle attacks, it was observed that all wireless communication was sent through the encrypted protocol HTTPS. This discovery indicates the camera's resistance to threat number 1 in Table IV. The lack of a local webserver reduces the attack surface of the device significantly. Thus, if this had been known before selecting this system to explore, a different system would probably have been selected instead for the sake of a larger attack surface.

```
       # hydra -L user_pass -P user_pass 192.168.0.85 telnet -V
[WARNING] telnet is by its nature unreliable to analyze, if possible better choose FTP, SSH, etc. if available
[DATA] max 16 tasks per 1 server, overall 16 tasks, 3721 login tries (l:61/p:61), ~233 tries per task
[DATA] attacking telnet://192.168.0.85:23/
[ATTEMPT] target 192.168.0.85 - login "Adminpldt" - pass "Adminpldt" - 1 of 3721 [child 0] (0/0)
[ATTEMPT] target 192.168.0.85 - login "Adminpldt" - pass "adminpldt" - 2 of 3721 [child 1] (0/0)
[ATTEMPT] target 192.168.0.85 - login "Adminpldt" - pass "1234567890" - 3 of 3721 [child 2] (0/0)
[ATTEMPT] target 192.168.0.85 - login "Adminpldt" - pass "xc3511" - 4 of 3721 [child 3] (0/0)
[ATTEMPT] target 192.168.0.85 - login "Adminpldt" - pass "vizxv" - 5 of 3721 [child 4] (0/0)
       ...
[ATTEMPT] target 192.168.0.85 - login "tech" - pass "admin" - 3711 of 3721 [child 11] (0/0)
[ATTEMPT] target 192.168.0.85 - login "tech" - pass "1111111" - 3712 of 3721 [child 8] (0/0)
[ATTEMPT] target 192.168.0.85 - login "tech" - pass "1234" - 3713 of 3721 [child 2] (0/0)
[ATTEMPT] target 192.168.0.85 - login "tech" - pass "12345" - 3714 of 3721 [child 0] (0/0)
[ATTEMPT] target 192.168.0.85 - login "tech" - pass "54321" - 3715 of 3721 [child 1] (0/0)
[ATTEMPT] target 192.168.0.85 - login "tech" - pass "123456" - 3716 of 3721 [child 13] (0/0)
[ATTEMPT] target 192.168.0.85 - login "tech" - pass "7ujMko0admin" - 3717 of 3721 [child 12] (0/0)
[ATTEMPT] target 192.168.0.85 - login "tech" - pass "1234" - 3718 of 3721 [child 0] (0/0)
[ATTEMPT] target 192.168.0.85 - login "tech" - pass "pass" - 3719 of 3721 [child 1] (0/0)
[ATTEMPT] target 192.168.0.85 - login "tech" - pass "meinsm" - 3720 of 3721 [child 9] (0/0)
[ATTEMPT] target 192.168.0.85 - login "tech" - pass "tech" - 3721 of 3721 [child 7] (0/0)
1 of 1 target completed, 0 valid password found
```

Fig. 7. Output from one of the dictionary attacks with Hydra

Considering the OWASP top 10 most critical security risks, the top three security risks can be ruled out due to the previously mentioned results [36]. No weak, guessable, or hard-coded passwords were discovered. No exploitable insecure network services are running on the device. The ecosystem interfaces are secure due to encryption.

Due to the time limit for the project, not all threats could be tested. Threat 1, 3 and 5 in Table IV were the ones possible to evaluate in this project due to partly the limited time, partly the level of competence of the project members in this area. Additional tests could be performed in order to evaluate some of the remaining threats. This is discussed in section X-A.

It can be noticed from the previous testing on SpotCam that found vulnerabilities have been remedied. The reason can be that security is taken seriously by the SpotCam corporation. The camera's simple design is also an indicator that more resources are put into security.

The fact that the vendors seem to prioritize security can contribute to the assessment that SpotCam Sense is a secure system. Other aspects that indicate that the system is secure are the lack of reported vulnerabilities found with the vulnerability scanners Nessus and MobSF.

A. Future work

Further attempts to deepen the security assessment and continue the penetration testing of SpotCam Sense can be done and more attacks can be found. For example attacks of relevance to the additional threats in the threat model.

The results from the analysis with MobSF enlightened the fact that clear text traffic is enabled for the mobile application. This was not discovered during the testing and can also be of relevance in future testing. The project did not include a physical examination of the device due to early delimitations. Therefore, examination of the camera's hardware will expand the assessment.

XI. Conclusions

The objective of this project was to assess the security level of the SpotCam Sense home security camera. Based on the penetration tests of this project, no exploitable vulnerabilities exist in the system. Because some delimitations were made, there are aspects of the security assessment that has not been covered in this project. Hence, a more thorough security assessment could be done on the system, which might result in a different conclusion. Applied countermeasures for the identified threats with the highest risk scores were discovered. Built on these findings and the assumption that the SpotCam developers previously proven to redeem security issues, SpotCam Sense home security camera is assessed to have sufficient safety precautions implemented against malicious hackers.

Appendix A
Username and password list for Hydra dictionary attack

Acknowledgment

We would like to thank our supervisor Pontus Johnson for his support during this project and for his contribution to arouse our interest in the subject of IoT security.

References

[1] J. Bugeja, B. Vogel, A. Jacobsson, and R. Varshney, "Iotsm: An end-to-end security model for iot ecosystems," in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Jun 2019, pp. 267–272.

[2] (2021, Apr) Spotcam. Bauhaus. [Online]. Available: https://www.bauhaus.se/media/pdf/0108557A_4.pdf

[3] G. Weidman, *Penetration Testing*, 1st ed. San Francisco, USA: No Starch Press, 2014.

[4] Riksdagsförvaltningen. (2014) Brottsbalk (1962:700) svensk författningssamling 1962:1962:700 t.o.m. sfs 2021:249. Brottsbalken 4 kap. 9c §. [Online]. Available: https://www.riksdagen.se/sv/dokument-lagar/dokument/svensk-forfattningssamling/brottsbalk-1962700_sfs-1962-700

[5] (2021, Apr.) Wireshark. [Online]. Available: https://www.wireshark.org/docs/wsug_html_chunked/

[6] (2021, Apr.) Aircrack-ng. [Online]. Available: https://www.aircrack-ng.org/

[7] (2021, Apr.) Ettercap. [Online]. Available: https://www.ettercap-project.org/
[8] (2021, Apr.) mitmproxy. [Online]. Available: https://www.mitmproxy.org/
[9] (2021, Apr.) Nmap. [Online]. Available: https://www.nmap.org/
[10] (2021, Apr.) Nessus. [Online]. Available: https://www.tenable.com/products/nessus
[11] (2021, Apr.) Hydra. [Online]. Available: https://tools.kali.org/password-attacks/hydra
[12] (2021, Apr.) Enjarify. [Online]. Available: https://github.com/google/enjarify
[13] (2021, Apr.) Jd-gui. [Online]. Available: http://java-decompiler.github.io/
[14] (2021, Apr.) Mobile security framework - mobsf. [Online]. Available: https://github.com/MobSF/Mobile-Security-Framework-MobSF
[15] (2016, Jun) Home security camera isn't secure. spotcam in the spotlight. [Online]. Available: https://www.pentestpartners.com/security-blog/home-security-camera-isnt-secure-spotcam-in-the-spotlight/
[16] H. Sneider. (2017, Apr) Spotcam sense pro review. [Online]. Available: https://the-gadgeteer.com/2017/04/24/spotcam-sense-pro-review/
[17] M. Stanislav and T. Beardsley. (2015, Sep) Hacking iot: A case study on baby monitor exposures and vulnerabilities. Rapid7. [Online]. Available: https://media.kasperskycontenthub.com/wp-content/uploads/sites/63/2015/11/21031739/Hacking-IoT-A-Case-Study-on-Baby-Monitor-Exposures-and-Vulnerabilities.pdf
[18] C. Sanders, *Practical Packet Analysis, 3E: Using Wireshark to Solve Real-World Network Problems*. San Francisco, CA: No Starch Press, 2017.
[19] J. Postel and J. Reynolds, "Telnet protocol specification," Internet Requests for Comments, RFC Editor, STD 8, May 1983. [Online]. Available: http://www.rfc-editor.org/rfc/rfc854.txt
[20] R. T. Fielding, J. Gettys, J. C. Mogul, H. F. Nielsen, L. Masinter, P. J. Leach, and T. Berners-Lee, "Hypertext transfer protocol – http/1.1," Internet Requests for Comments, RFC Editor, RFC 2616, June 1999. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2616.txt
[21] E. Rescorla, "Http over tls," Internet Requests for Comments, RFC Editor, RFC 2818, May 2000. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2818.txt
[22] T. Dierks and E. Rescorla, "The transport layer security (tls) protocol version 1.2," Internet Requests for Comments, RFC Editor, RFC 5246, August 2008. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5246.txt
[23] M. J. Ingeno, *Software architect's handbook : become a successful software architect by implementing effective architecture concepts*, 1st ed., Birmingham, UK, 2018.
[24] A. Guzman, *IoT penetration testing cookbook : identify vulnerabilities and secure your smart devices.*, 1st ed. Birmingham, UK: Packt, 2017.
[25] (2009, nov) The stride threat model. Microsoft Corporation. [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20)?redirectedfrom=MSDN
[26] J. Meier, A. Mackman, M. Dunner, S. Vasireddy, R. Escamilla, and A. Murukan. (2010, Jul) Threat modeling. Microsoft Corporation. [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff648644(v=pandp.10)#step-6-rate-the-threats
[27] ——. (2006, Jan) Threats and countermeasures. Microsoft Corporation. [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff648641(v=pandp.10)?redirectedfrom=MSDN&fbclid=IwAR2R_hHUs6Mwgku-r-KCGNto359USBwV5aaGLVmmDg8h9hTR93_mRCU0QLE#network-threats-and-countermeasures
[28] R. Baloch, *Ethical Hacking and Penetration Testing Guide*, 1st ed. Philadelphia, PA: CRC Press, 2015.
[29] (2021, Apr.) sslstrip. [Online]. Available: https://tools.kali.org/information-gathering/sslstrip
[30] Nmap.org, "Chapter 15. nmap reference guide," Apr 2021. [Online]. Available: https://nmap.org/book/man.html
[31] S. Jetty, *Network Scanning Cookbook*, 1st ed., Birmingham, UK, 2018.
[32] gmedian. (2021, Apr) File vulners. [Online]. Available: https://nmap.org/nsedoc/scripts/vulners.html
[33] D. Miessler. (2020, Jul) telnet-betterdefaultpasslist. [Online]. Available: https://github.com/danielmiessler/SecLists/blob/master/Passwords/Default-Credentials/telnet-betterdefaultpasslist.txt
[34] J. Vicente Vallejo. (2915, Jul) Hacking-busybox-control. [Online]. Available: https://github.com/vallejocc/Hacking-Busybox-Control/blob/master/routers_userpass.txt
[35] @zh4ck. (2015, Sep) How i hacked my ip camera, and found this backdoor account. [Online]. Available: https://jumpespjump.blogspot.com/2015/09/how-i-hacked-my-ip-camera-and-found.html
[36] OWASP. (2018) Owasp internet of things top 10 2018. [Online]. Available: https://owasp.org/www-pdf-archive/OWASP-IoT-Top-10-2018-final.pdf

Achieving Full Attack Coverage and Compiling Guidelines for enterpriseLang

Anton Hagelberg and Joshua Sadiq

***Abstract*—As the number of digital systems grows yearly, there is a need for good cyber security. Lack of such security can be attributed to the demand on resources or even knowledge. To fill this gap, tools such as enterpriseLang can be used by the end-user to find flaws within his system, which he can revise. This allows a user with inadequate knowledge of cyber security to create safer IT architecture. The authors of this paper took part in the development of enterpriseLang and its improvement. This was done by suggesting improvements based on certain design guidelines, as well as attempting to achieve 100% attack coverage and improving the defense coverage.**
The results show a coverage increase of 0.6% for a specific model's attack steps. Further more, we find that nearly 84.6% of the compiled guidelines are met, followed by 7.7% that were not fully met and a similar amount that were non-applicable to enterpriseLang. As the language is still in development, there remains much work that can improve it. A few suggestions would be to increase the attack coverage by 100%, increasing the defense coverage and improving enterpriseLang to fulfill the design guidelines, which would ultimately ease future projects within this domain.

***Sammanfattning*—Då antalet digitala system ständigt ökar gör även behovet för cybersäkerhet. Avsaknad av sådan säkerhet kan åläggas avsaknaden av kunskap och resurser. För att fylla detta gap utvecklas ständigt nya medel. Ett sådant är enterpriseLang som kan användas av en utvecklare för att hitta säkerhetsbrister i sitt system. Detta tillåter en utvecklare med låg kunskap inom cybersäkerhet att utveckla säkrare system, applikationer och produkter. Skribenterna av denna avhandling tog del i utvecklingen och förbättringen av enterpriseLang. Detta gjordes genom att föreslå förbättringar baserade på särskilda designriktlinjer och ett försök att uppnå 100 % täckning av attack-steg.**
Resultaten visar på en ökning av 0.6% anfallstäckning för en specifik modell. Vidare visar de att 84.6 % av riktlinjerna är uppfyllda, 7.7% var inte uppfyllda och 7.7% var inte relevanta för enterpriseLang. Då språket ännu är i utvecklingsstadiet finns ännu mycket arbete kvar att göra. Några förslag är att öka anfallstäckningen till 100%, öka försvarstäckningen samt förbättra språket så den når upp till alla designriktlinjer.

***Index Terms*—Threat Modeling, Cyber Security, Domain Specific Language, Meta Attack Language, enterpriseLang**

Supervisors: Robert Lagerström, Simon Hacks and Wenjun Xiong

TRITA number: TRITA-EECS-EX-2021:183

I. Introduction

A. Background

With an increasing amount of digital systems every year, there is an evergrowing need for good cyber security. Good cyber security ensures the safety of vital systems and integrity of individuals, as well as the well-being of companies, governments and nations [1].

As a digital system is developed, there can be many reasons why the security is lacking, or outright flawed. Just as the rest of the system, security requires resources such as time and money. Furthermore, cyber security is its own vast field that may or may not be part of the users skillset, spawning an obstacle as an inadequacy in knowledge appears. In the aforementioned situation, the Meta Attack Language - henceforth referred to as MAL - can be of usage [2].

The MAL produces a formalism which allows the creation of Domain-Specific Languages (DSLs) within the domain of threat modeling *and* attack simulations. By providing the necessary foundation and guidelines, it is possible to create DSLs which can be crucial in the process of threat modeling and simulating cyber attacks. A DSL can be utilized to describe a system, its assets, defenses and the relation between all parts [2]. Once such a system is described, generating a number of virtual penetration tests - cyber attacks - becomes much easier as describing each subsystem becomes obsolete.

One such DSL is enterpriseLang [3], currently in development at the Royal Institute of Technology, Sweden (KTH). In situations where large number of systems are used within an enterprise, enterpriseLang becomes a tool to address possible security weaknesses [3].

As part of the authors' Bachelor theses, the purpose of this project is to partake in the development and improvement of enterpriseLang. There is currently a substantial amount of attack as well as defense coverage missing in regards to the language. By increasing the coverage, the correct functionality of the DSL can be ensured when running attack simulations. Furthermore, the DSL has yet to be put under scrutiny by design guidelines. These crucial improvements are given to the project group to handle.

B. Goals

The goals of this project are to attain a 100% attack coverage, increasing the defense coverage and comparing the DSL to certain guidelines. Achieving 100% attack coverage will allow the language to be validated, as all attack steps are covered. The same principle applies to increased defense coverage. DSLs are created using a few sets of guidelines, outlined by G. Karsai et al. [4]. In order for enterpriseLang to reach up to its full potential it is important for these guidelines to be, at the very least, partially met as outlined in IV.

II. Background

A. Cyber security and the Meta Attack Language

As the term *cyber security* continues to grow in usage [5], there is a necessity to clarify the definition of the term. Schatz et al. [6] achieve this by conducting an analysis of the literature and providing an improved definition, namely:

> The approach and actions associated with security risk management processes followed by organizations and states to protect confidentiality, integrity and availability of data and assets used in cyber space. The concept includes guidelines, policies and collections of safeguards, technologies, tools and training to provide the best protection for the state of the cyber environment and its users [6].

To this collection of tools and technologies belongs the aforementioned Meta Attack Language. To explain how MAL works, it is more convenient to understand why it is needed altogether.

An attack simulation is a useful tool in assessing the potential weaknesses of a system. By simulating possible paths of attacks that a hacker could choose to take, it is possible to find the estimated time it takes for the attacker to compromise an asset of interest, i.e. gain access to the asset. To visualize this process, attack graphs are often utilized [2]. An attack graph can consist of various nodes that represent states of networks as well as 'edges' that represent the paths between these nodes [7]. In Fig. 1, an example of a simplistic attack graph is portrayed. The hacker chooses his entry point to be a browser, i.e. exploiting a security weakness or flaw in the browser to gain entry into the system. Through this, he can enter the computer service, followed by the OS and finally the user account. Once at this point, he could have partial or full control over the computer.

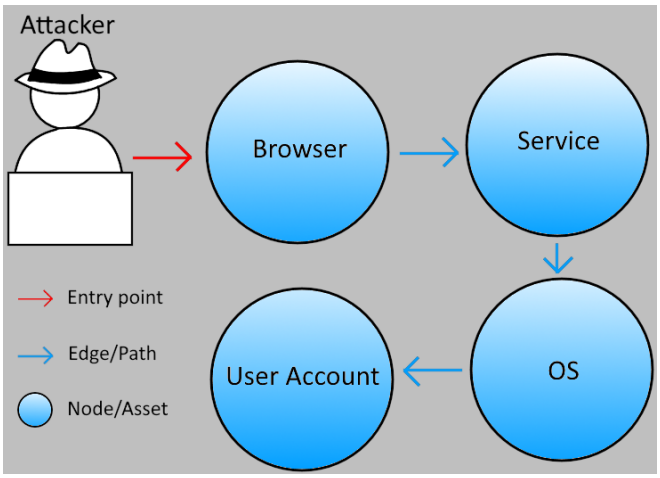


Fig. 1. An example of a simple attack graph.

As there exist many various forms of systems, employing attack simulations require new attack graphs to be built for each system, which is costly and time-consuming. To resolve this issue, it is possible to construct 'domain-specific languages' which can be reused for systems within the same domain. An example of such a domain could be the embedded systems within a car. MAL provides a formalism which allows designing DSLs for this purpose [2].

Developed at KTH in collaboration with experts from Foreseeti AB[1], MAL is a specification language that provides guidelines for the creation of DSLs. MAL provides the option of taking probability distribution into account when calculating the estimated time until the asset of interest is compromised. Since probability distribution is optional to the attack simulation and not necessary when increasing attack or defense coverage, the scope of this project does not cover it. Furthermore, MAL can compute large attack graphs quite efficiently [2].

B. enterpriseLang

To date, MAL has been used in creating several various DSLs, with their respective purposes. coreLang, developed by KTH and Foreseeti, excels in the IT domain [8]. vehicleLang, developed by researchers at KTH and RWTH Aachen University, is another example of a DSL created with MAL as its specification, and is used for attack simulations in the IT systems of vehicles [9]. A final example would be powerLang, a DSL used in the power domain, also developed at KTH [10]. As the project group has been given the task of improving enterpriseLang, which is currently being developed at KTH by doctoral student Wenjun Xiong, the focus of this paper will be on it alone. Despite this, the process of increasing coverage and the resulting design guidelines can easily be applied to other MAL-based DSLs as they share the same organizational structure, purpose and properties.

Much like the aforementioned DSLs developed with the MAL-specification, enterpriseLang allows the user to simulate cyber attacks on their system, with the optional capability to simulate probabilistic Time-to-Compromise (TTC). enterpriseLang is designed to be used in attack simulations with enterprise systems where cloud services, workstations and mobile systems are combined [3].

At the time of writing, several components of enterpriseLang require further development and validation such as increased coverage and applying additional probabilities to various attack steps and defense entities. Elements central to this paper are attack coverage, defense coverage and the language's adaption to design guidelines, as the authors seek to provide a qualitative structure on which core functionality can be evaluated and to what degree the language adheres to guidelines.

C. Coverage

When software is developed, testing is a necessary part of the process to ensure software bugs are found and debugged. *Test coverage* refers to the percentage of a software's source code that is executed when test suites are run. High test coverage would signify that the chance of bugs to appear is low, as larger parts of the software was executed during the test run [11]. Similarly, *attack* and *defense coverage* refers to the percentage of attack paths respectively defense entities

[1]https://foreseeti.com/

that are covered or activated in an attack simulation [12]. Achieving full attack and defense coverage ensures that a DSL is validated and reliable to be used when conducting attack simulations. This is done by producing test suites that activate all paths and entities for models within the DSL.

D. Implementation

1) Prerequisites and Tools: To achieve the necessary increase in attack coverage as well as defense coverage, it is crucial to be acquainted with the tools used in this process. Prerequisites to run the tools and develop the language are:

- **Java 14**
  Java is needed to run most tools used in the process. Java 14 is needed to run the MAL Coverage Viewer.
- **Java JDK**[2]
  The Java JDK contains the tools necessary to develop Java software. Most importantly, the JDK contains the Java Compiler, which is a necessity when compiling test files.
- **Apache Maven**[3]
  Apache Maven is a build automation tool. It handles how software is built as well as dependencies.
- **Code editor**
  A simple one will suffice, such as Notepad++.

Furthermore, there is a number of tools that are necessary in the process of increasing the coverage:

- **MAL Compiler**[4]
  This tool allows the compilation of Java test files with a DSL's MAL-specification file as backend reference.
- **Modified MAL Compiler**[5]
  This tool is a fork of the aforementioned compiler, modified by Nicklas Hersén. While this compiler also compiles test files with the MAL-specification as a reference backend, it also allows the generation of coverage-related Java-class files, which produce coverage metric when an attack simulation is run.
- **JUnit 5 Standalone Launcher**[6]
  This tool will run our attack simulations on the compiled test classes and produce a file with coverage-related data, known as a JSON-file.
- **MAL Coverage Viewer**[7]
  By feeding it the JSON-file generated from the JUnit compilation, it is possible to view all test models, as specified in the test files, depicted in Fig. 2. Furthermore, the tool allows us to see various coverage-related metric for each test case, as can be seen in Fig. 3. Unfortunately, this tool does not calculate aggregated coverage data, i.e. it does not show whether different test cases of the same model overlap in coverage. This complicates the process of achieving *full* attack coverage.

[2]Java SE
[3]Apache Maven
[4]https://github.com/mal-lang/malcompiler
[5]https://github.com/nicklashersen/malcompiler
[6]JUnit 5
[7]https://github.com/nicklashersen/mal-coverage-viewer

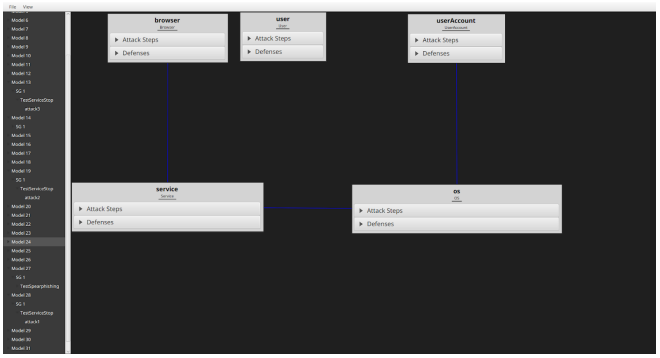


Fig. 2. MAL Coverage Viewer in action, visualizing a specific test model, its assets and the steps between them.

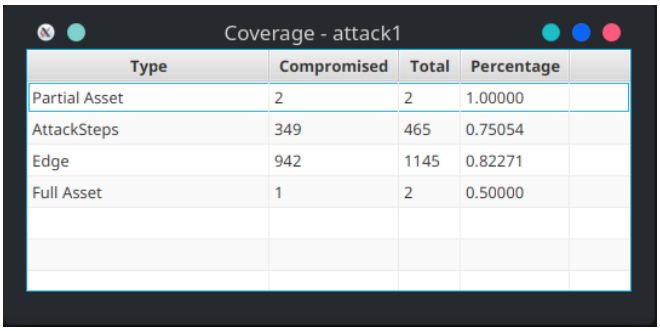


Fig. 3. MAL Coverage Viewer displaying coverage metric for a specific test case titled 'attack1'.

2) Extending the Test Classes With CoverageExtension: To generate the JSON-file with coverage-related data, it is necessary to add coverage extensions for every file, as well as registering each class with a global exportable target:

```
import org.junit.jupiter.api.extension.
RegisterExtension;
import org.junit.jupiter.api.extension.
ExtendWith;
import core.coverage.JSONTarget;
import core.coverage.ConsoleTarget;
import core.coverage.CoverageExtension;

@ExtendWith(CoverageExtension.class)
```

Following this step, it is possible to compile the test files with the modified MAL Compiler.

3) Finding Missing Coverage: Once the compiler has produced a set of test classes, it is required to link them against the standalone JUnit launcher. Once done, it is possible to run the JUnit.jar-file, which produces a coverage.json-file. This file can then be used in conjunction with the Coverage Viewer to illustrate the various models and their test cases.

Coverage-related data, as shown in Fig. 3, can be found under *View* > *Coverage Info*. From here, it is possible to see what partial assets, attack steps, edges, defense and full assets are compromised. At the very right, a percentage is calculated and shown. *AttackSteps* and *Defense* were of particular interest for this project.

To find the compromised attack steps and defense entities,

it is merely a matter of opening the dropdown box for a specific asset. Text in red signifies compromised steps or entities, whereas black ones are uncompromised. Furthermore, attack steps with the &-prefix denote an AND-step. Likewise, | show an OR-step. Finally, #-prefix denote the defense entities. Finding missing coverage is a matter of finding steps or entities in black.

As mentioned earlier, this tool does not calculate aggregated coverage nor does it portray it in the models it illustrate. As each model is generated from respective test case, it is possible that different attack steps are covered in various test cases, uniting to produce full coverage for the same model. Likewise, it is possible that various test cases produce the same attack steps, and ultimately unite into a less-than-full coverage.

E. Guidelines

Since enterpriseLang was developed in accordance with the Design-Science-Research (DSR) objectives outlined by Peffers et al. [13], the design guidelines outlined by Karsai et al. [4] and Kahraman et al. [14], we wish to evaluate exactly how these proposed guidelines were followed. If they were not followed, we wish to propose changes which would be more in line with the guidelines.

III. Method

A. Six Objectives

Proposed by Peffers et al. [13] we can split the evaluation process into six objectives.

Objective list	
Objective	Measure
1. Problem identification	Important and relevant problems
2. Objectives of the solution	Implicit in ”relevance”
3. Design and development	Iterative search process
4. Demonstration	Simulation, experiments
5. Evaluation	Evaluate
6. Communication	Communicating

1) Problem identification: This objective refers to identifying a problem and has seen quite some development, focusing on programming and data collection to identifying a need by Peffers et al. [15]. In this paper the objective was realized before we started working, as our work has been focused on improving an already realized language, namely enterpriseLang.

2) Objectives of the solution: By inferring the objectives of a solution from the problem definition the reader can achieve a sense of why the current work trumps previous work and is therefore necessary. Since enterpriseLang covers a completely new domain it can be argued that there is no previous work to compare it to, making it better by default. To ensure the correct functionality of these MAL-based languages e.g. enterpriseLang, a qualitative and quantitative assessment is needed which this paper aims to provide.

3) Design and development: Create the solution. This work has mostly been finished by the development of enterpriseLang by Wenjun Xiong [3], but this project aims to extend upon that by the work covered in IV. Since we wish to increase the attack coverage of models it this is the first objective we work with directly in this paper.

4) Demonstration: A big part of the work has been to increase both attack coverage and evaluate the set of guidelines. With the demonstrative objective being focused on actual experimentation and simulations, the process of extending attack coverage makes full use of demonstration. In this paper we demonstrate both how to increase attack coverage and evaluate guidelines, whereby the results can be found in IV.

5) Evaluation: In this paper, a large part of the work has been focused around evaluation. This means that the main goal was to observe and measure how well the language supports the expected solution to the problem stated. Once the evaluation stage has been finished, we can then iterate back to step 3 in order to improve the language and fix the flaws which can appear during the evaluation objective.

6) Communication: This report attempts to communicate the problem itself. It is important to realize that in this report, an extension of the solution originally provided by enterpriseLang is also formulated.

B. Attack Coverage

As mentioned in I-B, part of the work was to achieve 100% attack coverage and to increase the defense coverage.The process of increasing the coverage will be detailed below.

To increase the coverage, new test cases have to be written. These test cases require careful consideration of what attack steps are covered in a test case and other cases of the same model. By finding uncompromised attack steps with the use of the Coverage Viewer, it is possible to use them in new test cases to increase the coverage by asserting them as compromised. Practically, this would translate into the hacker's entry point were he to attack the system which the model is made after.

Despite the straightforward method, picking an attack step as a hacker's entry point when designing a new test case is not merely a matter of finding uncompromised ones. A realistic scenario has to be portrayed in the developer's mind when doing so. For example, a hacker would not be able to choose an OS's kernel as their entry point. Choosing a spearphishing email attachment, a browser extension or even perhaps a physical workstation as the point of entry would make more sense, as such attacks have been conducted in the past and are viable. In this work, assets that were taken into consideration were either those with network connection outside the system, or assets with edges to already-compromised assets.

C. Guidelines

When evaluating guidelines the process is divided into two dimensions, as described by J. Venable [16].

- Dimension 1: Functional Purpose of the Evaluation
- Dimension 2: Paradigm of the Evaluation Study

The first dimension refers to why we want to evaluate the guidelines in the first place. DSLs are complex languages with many elements to consider whilst designing. As outlined in IV, some of these guidelines refer to the code itself in terms of simplicity, readability and comprehensibility. Everyone involved with the DSL greatly benefit from these guidelines being taken into consideration during the design phase. However, it is not always possible to take all different guidelines into account. This ties into the second dimension; how do we evaluate these guidelines? In simple terms, it is a matter of identifying the guidelines requirements for fulfillment and checking whether the current version of the language meets these requirements.

IV. Results

A. Attack and Defense Coverage

At the time of writing, a 0.6% increase for attack steps and 0.3% increase for attack edges in the ServiceStop-model were achieved through the addition of a single test case. This model assumes that the attacker enters through the Windows-OS `attemptServiceStop`. In the situation where no defenses are active, 75.1% of the attack steps were compromised. In our test case, we assume an additional entry point through the Windows Control Panel, which increased the compromised attack steps to 75.5%. This increase measures to 0.6% for attack steps, as mentioned earlier. As for attack edges, the percentage increased from 82.3% to 82.5%, a gain of 0.3%. These results are depicted below. In Fig. 4 we can observe the coverage metrics before our addition. In Fig. 5, we find the updated coverage metrics with our additional entry point.

Coverage - attack1

Type	Compromised	Total	Percentage
Partial Asset	2	2	1.00000
AttackSteps	349	465	0.75054
Edge	942	1145	0.82271
Full Asset	1	2	0.50000

Fig. 4. Coverage data for the ServiceStop-model with no defenses prior to increase.

In regards to defense coverage, no feasible results were attained.

B. Guidelines

The guidelines below are a compilation of a multitude of various guidelines from two separate papers. A majority come from G. Karsai et al. [4] and G. Kahraman & S. Bilgen [14]. Notably, 3 of the guidelines from the different papers relate to one another and can be favorably combined. Do note that these are simply an extract of the many different guidelines, thus not all the guidelines in the aforementioned papers are noted below. In Fig. 6 we can see a piechart depicting the ratio between the status of various guidelines.

Coverage - attack3

Type	Compromised	Total	Percentage
Partial Asset	2	2	1.00000
AttackSteps	351	465	0.75484
Edge	945	1145	0.82533
Full Asset	1	2	0.50000

Fig. 5. Coverage data for the ServiceStop-model with no defenses post-increase.

1) Fully met: **The language's purpose should be clear from the get-go, whether that is one specific usage or a variety of situations.**
This is one example of a guideline which enterpriseLang adequately lives up to. The purpose of the language is made clear both in documentation on the GitHub page[8] and through examination of the files present.
Is the DSL adequately simplistic?
From an outsider's point of view the concept of a DSL is a foreign one. The documentation does however sufficiently describe how the language is built and what the focus is. On the GitHub project page there is an extensive collection of probability distributions which adequately describes different types of attacks. The concept of simplicity is however a questionable one: Is it relevant to consider simplicity when creating a new DSL? Or is simplicity interpreted as being 'simple to use'? In this regard we choose to interpret this as being simple to use.
For an experienced user of Meta Attack Languages, using enterpriseLang will be easier than for someone who has not worked with one before. If someone with no programming experience tries to understand enterpriseLang it will come with some hardship. However, expecting the user to have at the very minimum prior programming experience, we can consider this guideline met.

2) Partially/Not fully met: Some guidelines are partially or not fully met, meaning that there remains future work which can improve the language as a whole.
Provide organizational structures for models.
This guideline is, at the time of writing, not met as the organizational structures are all stored in a single file.
Is the project structure clear?
Who is modeling in the DSL? Who is reviewing the models? These are all questions that need to be clarified.
Who the DSL is created for is not obvious. The fact that it is designed to assess the security of systems within an enterprise is apparent, but who models and who reviews the models is currently up to interpretation.

3) Non-applicable: As mentioned in III-C, some guidelines cannot be fulfilled or simply are not relevant, meaning the guideline itself is not applicable. Some examples of these are:
Using syntactic sugar appropriately.

[8]https://github.com/KTH-SSAS/enterpriseLang

Syntactic sugar refers to elements of the language which do not contribute to expressiveness but rather only serve to increase readability. Since syntactic sugar is defined on the MAL level of the language and is not defined on a sub-language level such as a DSL, this guideline is not relevant to enterpriseLang istelf.

Minimizing development time and human resources. We choose to interpret this guidelines as something which should be done going forward. But seeing as a majority of enterpriseLang is already fully developed it becomes less relevant to consider.

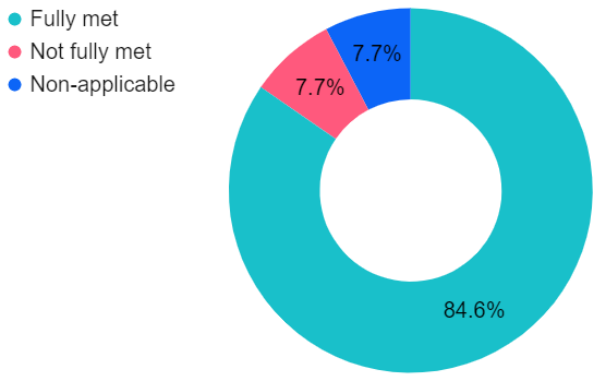


Fig. 6. Piechart depicting the ratio between fully met, not fully met and non-applicable guidelines.

V. Discussion

A. Analysis of Coverage Results

The project's initial goal was to achieve full attack coverage and increase the coverage of defense entities. Due to unforeseen circumstances, progress was slower than expected. Many of these complications are detailed further below. One unexpected issue was the lack of aggregated coverage that the MAL Coverage Viewer displays. As this was uncovered, the goal shifted from achieving full attack coverage to increasing the attack coverage as much as possible for each model. Calculating the aggregated coverage percentage is possible, but due to the lack of time it was not considered. Due to time constraints, this only led to the increase of a single model (detailed in IV) by 0.6% for attack steps and 0.3% for attack edges.

B. Guidelines

As can be seen in Fig. 6. a total of 84.6% of the guidelines proposed by G. Karsai et al. [4] and G. Kahraman & S. Bilgen [14] can be considered fully met. However that leaves us with 7.7% that are not currently met. How shall these guidelines be fulfilled? One aspect would be to create some form of framework which would support a one-file-per-asset system, instead of the current system where all assets are in the same file. This would satisfy the criteria for providing organizational structure for the models.

Clarity regarding who is meant to model in the DSL is, as mentioned in IV, a guideline which is not currently met. This guideline is rather obscure as DSLs and MALs have a small amount of users, however we still believe it is important, which is why it is considered relevant. The problem could be solved rather easily, either in the documentation or through the GitHub page, by providing some form of commentary regarding who the language is built for.

C. Complications

As presented in IV, the achieved increase for attack steps was a mere 0.6% for the ServiceStop-model. There are many reasons why the results fall short of achieving full attack coverage as well as increasing the defense coverage.

The main factor that contributed to our low increase was the difficulty of understanding the tools used in this process. Our experience show that Windows-based workstations can produce significantly more issues when using the tools, particularly if the user is not experienced in this field, and that Linux-based machines provide much more ease-of-use.

Furthermore, a clear distinction needs to be made between the MAL Compiler and the modified version. The latter is capable of compiling test files with additional CoverageExtensions and generating coverage-related classes. These are, in turn, necessary to generate the coverage file.

Finally, due to lacking documentation it was not apparent that the specific software versions of Java and Gradle the MAL Coverage Viewer required were Java 14 and Gradle 7.0.

As the field of cyber security was fairly new to the authors, finding realistic and viable entry points when writing new test cases was a challenge. Due to the lack of experience and knowledge, it was not apparent what entry points as well as attack paths a hacker could utilize. This led to somewhat uncertain assertions to be made.

D. Reflection

As neither of the authors had previously worked with cyber security, this field was new, exciting and challenging to engage with. The terminology, tools, methodology and thought process used in this project were all unfamiliar and required much time to learn. Considerable knowledge was attained regarding the process of DSR, test coverage, cyber security, and other related subjects. By the end of this project, the authors stand more confident to work with related subjects.

E. Future Work

As achieving full attack coverage was not reached in this project, one task for future work would be to increase the coverage for attack steps and attack edges to a 100%, for all current models. As the implementation and methodology of doing so are clearly outlined in this report, we propose for researchers new to this field to carefully read through those sections.

Another task closely related to this is the increase of defense coverage. By achieving full defense coverage, the reliability of the language can be validated. The methodology for doing this is very similar to increasing the attack coverage and therefore also outlined by this report.

Something we found when going through the MAL-specification file was that not all attack steps and defense

entities have a probability distribution applied. A third task would be to apply probability distribution to all steps and entities, thereby fully validating the option to use statistical determination during attack simulations. As this was not within the scope of this project, we refer to Johnson et al. [2].

Our work resulted in a compiled list of design guidelines which pinpointed what aspects of enterpriseLang were not fully met. One future assignment is to improve the elements of the language which will result in the language fully meeting the design guidelines. This will ensure that the language is more reliable, as it reaches up to a set standard.

As this work outlines how to implement and produce new test cases that increase attack and defense coverage for MAL-based DSLs, future DSLs could refer to this work in the developmental phase. As MAL-based DSLs share a similar structure, applying these methods would work quite well. With the same reasoning, applying our design guidelines to other MAL-based DSLs (and potentially new ones) would be beneficial to the improvement and development of these languages.

Finally, many of the tools used in this project could improve. One particular suggestion for improvement would be adding the option to view aggregated coverage data in the MAL Coverage Viewer. At the moment, this can be done although it is an arduous task. With the option to view aggregated coverage, the task of increasing coverage would become significantly easier.

VI. Conclusions

In short, the authors of this report set out to partake in the development of enterpriseLang by increasing coverage for attack steps and defense entities as well as validating the language to design guidelines. Having achieved some of these results, enterpriseLang is one step closer to be used by developers for the purpose of improving the cyber security in their products.

Appendix 1 - DSL Design Guidelines for enterpriseLang Purposes

Appendix 2 - Improved TestServiceStop

VII. Acknowledgements

We want to acknowledge the support and help from our project supervisors Simon Hacks and Wenjun Xiong as well as our academic supervisor Robert Lagerström. Furthermore, we appreciate the quick replies to our emails from course supervisor Anita Kullen.

References

[1] J. Wakefield. (2017, june) Tax software blamed for cyber-attack spread. Broadcasting House, London, England. [Online]. Available: https://www.bbc.com/news/technology-40428967

[2] P. Johnson, R. Lagerström, and M. Ekstedt, "A meta language for threat modeling and attack simulations," in *Proceedings of the 13th International Conference on Availability, Reliability and Security*, ser. ARES 2018. New York, NY, USA: Association for Computing Machinery, 2018. [Online]. Available: https://doi.org/10.1145/3230833.3232799

[3] W. Xiong, E. Legrand, O. Åberg, and R. Lagerstrom, "Cyber security threat modeling based on the MITRE Enterprise ATT&CK Matrix," *submitted*, 2020.

[4] G. Karsai, H. Krahn, C. Pinkernell, B. Rumpe, M. Schindler, and S. Völkel, "Design guidelines for domain specific languages," *CoRR*, vol. abs/1409.2378, 2014. [Online]. Available: http://arxiv.org/abs/1409.2378

[5] trends.google.com, "Google trends: Cybersecurity," May 2021. [Online]. Available: https://trends.google.com/trends/explore?date=all&q=cybersecurity

[6] D. Schats, R. Bashroush, and J. Wall, "Towards a more representative definition of cyber security," *Journal of Digital Forensics, Security and Law*, vol. 12, 06 2014.

[7] X. Cyber. What are attack graphs? Herzliya, Israel. [Online]. Available: https://www.xmcyber.com/attack-graphs/

[8] S. Katsikeas, S. Hacks, P. Johnson, M. Ekstedt, R. Lagerström, J. Jacobsson, M. Wällstedt, and P. Eliasson, "A probabilistic attack simulation language for the it domain," ser. GraMSec 2020, vol. 12419, 2020.

[9] S. Katsikeas, P. Johnson, S. Hacks, and R. Lagerström, "Probabilistic modeling and simulation of vehicular cyber attacks: An application of the meta attack language." in *ICISSP*, 2019, pp. 175–182.

[10] S. Hacks, S. Katsikeas, E. Ling, R. Lagerström, and M. Ekstedt, "powerlang: a probabilistic attack simulation language for the power domain," *Energy Informatics*, vol. 3, no. 1, pp. 1–17, 2020.

[11] L. Brader, H. Hilliker, and A. Wills, "Testing for continuous delivery with visual studio 2012," *Microsoft*, p. 30, 2013.

[12] N. Hersén, S. Hacks, and K. Fögen, "Towards measuring test coverage of attack simulations," in *Proc. of the 25th International Conference, EMMSAD 2021, Held at CAiSE 2021 (to be published)*, 2021, pp. 1–15.

[13] K. Peffers, T. Tuunanen, C. Gengler, and M. Rossi, "The design science research process: a model for producing and presenting information systems research," *Proceedings Design Research Information Systems and Technology DESRIST'06*, vol. 24, 01 2006.

[14] G. Kahraman and S. Bilgen, "A framework for qualitative assessment of domain-specific languages," *Software & Systems Modeling*, vol. 14, no. 4, pp. 1505–1526, Oct 2015. [Online]. Available: https://doi.org/10.1007/s10270-013-0387-8

[15] K. Peffers, Ken, T. Tuunanen, Tuure, M. Rothenberger, Marcus, Chatterjee, and Samir, "A design science research methodology for information systems research," *Journal of Management Information Systems*, vol. 24, pp. 45–78, 01 2008.

[16] J. Venable, J. Pries-Heje, and R. Baskerville, "Feds: a framework for evaluation in design science research," *European Journal of Information Systems*, vol. 25, no. 1, pp. 77–89, 2016. [Online]. Available: https://doi.org/10.1057/ejis.2014.36

Evasion Attacks Against Behavioral Biometric Continuous Authentication Using a Generative Adversarial Network

Erik Sävenäs and Herman Blenneros

***Abstract*—The aim of the project was to examine the feasibility of evading continuous authentication systems with a generative adversarial network. To this end, a group of supervised and unsupervised state-of-the-art classifiers were trained on a publicly available dataset of stroke patterns on mobile devices. To find the best configurations for each classifier, hyper-parameter searches were performed. To attack the classifiers, a generative adversarial network was trained on the dataset to reproduce samples following the same distribution. The generative adversarial network was optimized to maximize the Equal Error Rate metric of the classifiers on the reproduced data. Our results show that the Equal Error Rate and the Threshold False Acceptance Rate increased on generated samples compared to random evasion attacks. Across the classifiers, the greatest increase in Equal Error Rate was 26 percent (for the artificial neural network), and the greatest increase in Threshold False Acceptance Rate was 60 percent for the same classifier. Moreover, it was found that, in general, the unsupervised classifiers were more robust towards this type of attack. The results indicate that evasion attacks against continuous authentication systems using a generative adversarial network are feasible and thus constitute a real threat.**

***Sammanfattning*—Målet med detta projekt var att undersöka möjligheten att undgå ett aktivt verifieringssystem med hjälp av ett generativt nätverk. För att göra detta valde vi ut ett antal moderna klassifieringsalgoritmer och tränade dem på en offentlig datasamling av svepmönster på mobiltelefoner. För att erhålla de bästa konfigurationerna för varje klassifieringsalgoritm utfördes hyper-parameter sökningar. För att attackera klassifieringsalgorithmerna implementerades ett generative adversarial network som tränades på datasamlingen för att reproducera liknande svepmönster. Det generativa nätverket optimerades för att maximera klassifieringsalgoritmernas likvärdiga felkvot med den reproducerade datan. Resultaten visar att den likvärdiga felkvoten och tröskeln av den felaktiga verifieringskvoten ökade med den reproducerade datan jämfört med slumpmässiga tester. Den högsta ökningen av den likvärdiga felkvoten var 26 procent (för det artificiella neurala nätverket) och den högsta ökningen av tröskeln av den felaktiga verifieringskvoten var 60 procent for samma algoritm. Därutöver fann vi att de oövervakade klassifieringsalgoritmerna var mer motståndskraftiga mot denna typen av attack jämfört med de övervakade klassifieringsalgoritmerna. Resultaten tyder på att det är möjligt att till viss del undgå ett aktivt verifieringssystem med hjälp av ett generative adversarial network och att denna typen av attacker utgör ett konkret hot.**

***Index Terms*—Continuous Authentication, Generative Adversarial Network, evasion, biometrics.**

Supervisors: Ezzeldin Zaki

TRITA number: TRITA-EECS-EX-2021:183

I. Introduction

Traditional authentication systems have several security flaws when it comes to user verification. For example, spoofing [1], smudge attacks [2], and guessing [3] are forms of attacks to which these systems are susceptible. One potential solution to deal with such attacks is the emerging concept of continuous authentication (CA).

In recent years, several CA frameworks based on different biometric data have been proposed. Systems associated with mouse dynamics [4] and lip-movement [5] are some examples. Another popular biometric for active user authentication is touchscreen dynamics [6], [7] and [8]. In [6], user touch data was recorded by raw touchscreen logs of a standard API of mobile devices and used to train algorithms capable of classifying data. Similarly, the authors in [7] proposed a CA solution based on the same type of data. However, a user worn sensor-glove was used to cross-validate the classifications made by the system. The authors in [8] proposed a framework combining hand movement, phone orientation and grasp for continuous user authentication.

While the advancements in the field of CA seem promising, and good performances have been achieved by proposed solutions, the results have been achieved by random evasion attacks against the classifiers. This means that the tests have been conducted by presenting random samples to the classifiers which is comparable to a user trying to access another user's device by using their own biometric data. Little work has been done towards strengthening CA frameworks against more sophisticated attacks. In [9], two methods for fooling a classifier were proposed: one numerical method and one method based on randomization. The authors concluded that such attacks were feasible with high success rates which suggests that the security of CA systems should be put under scrutiny.

One potential approach to evade CA systems is by using generative modeling. Deep models such as a Generative Adversarial Networks (GANs) have gained popularity in recent years for their ability to reproduce data following the same distribution. In [10], a neural network was used to generate words and phrases spoken by humans, as well as pieces of music. Perhaps closest to the work made in this project, the authors in [11] trained a GAN on human fingerprints and used a stochastic search method for the input to the generator network that produces the highest amount of false acceptances by finger recognizers.

In this project, we explore the feasibility of evading continuous authentication systems by mimicking authenticated users using a Generative Adversarial Network. To this end, we train a group of state-of-the-art classifiers to authenticate users based on their touchscreen stroke patterns from a publicly available dataset [12]. A Generative Adversarial Network is trained to reproduce the same data and the reproduced samples are used in an attempt to evade the classifiers.

II. Background

In this section we provide an overview of the concepts essential to this project. In II-A, we give a presentation of continuous authentication and distinguish between biological and behavioral biometric data. In II-B, we explain the concept of classification and give a brief notion of what machine learning is. We also explain the difference between supervised and unsupervised learning. Lastly, we go through the basics of artificial neural networks in II-C and Generative Adversarial Networks in II-D.

A. Continuous Authentication

Continuous authentication has to do with passively authenticating the user of a device. A CA system does not demand a phrase or string of characters such as a password for user differentiation. In contrast to traditional one-time authentication, CA frameworks periodically assess a user's credibility and adjusts the user's privileges accordingly. This implies that an ideal CA system is more robust towards session hijacking than traditional authentication schemes, including multi-factor authentication.

Continuous authentication systems are usually based on biometric data. Such data can generally be divided into two groups: *biological* biometric data and *behavioral* biometric data. Biological data contains information about personal traits connected to one's physical appearance. Facial features and fingerprints are both examples of such information. In contrast, behavioral data consist of information related to one's behavioral traits such as scrolling speed or the pressure someone exerts on their touchscreen when dialing a phone number. In this project we will focus on behavioral data.

B. Classification

The software component of CA used to decide whether a user should be allowed continued access to a device is called a classifier. In general, a classifier is anything capable of categorizing data. In this project, the classifiers will be based on machine learning, meaning that they are computer algorithms capable of improving automatically from training on data. For authentication purposes, *binary* classification is most widely used. A binary classifier can categorize data into two classes (e.g., authorized or unauthorized). In this project, five binary classifiers of two different types were used: three *supervised* classifiers and two *unsupervised* classifiers. The terms *supervised* and *unsupervised* refer to the way the machine learning models are trained. Supervised learning is a method used to train a machine learning algorithm which involves providing the algorithm with information about the existing categories. In practice, this means coupling the samples with their correct labels (i.e., their categories). In contrast, unsupervised learning is a method used to train a machine learning algorithm where the model is not provided any labels and instead has to group the data based on various statistical methods.

C. Artificial Neural Network

Artificial Neural Networks (ANN), also known simply as neural networks, are computing algorithms inspired by the human brain. Their architecture consists of layers of neurons: one input layer, one or more hidden layers, and one output layer. Between each pair of neurons is a connection weighted according to its importance. See Fig 1 for the structure of a neural network.

At the ith neuron of a layer, the output y_i is calculated by

$$y_i = A\big(\sum_j w_j x_j + b_i\big), \tag{1}$$

where A is the activation function of the layer, w_j is the weight associated to the jth connection to the neuron, x_j is the jth input to the neuron, and b_i is the neuron's bias term.

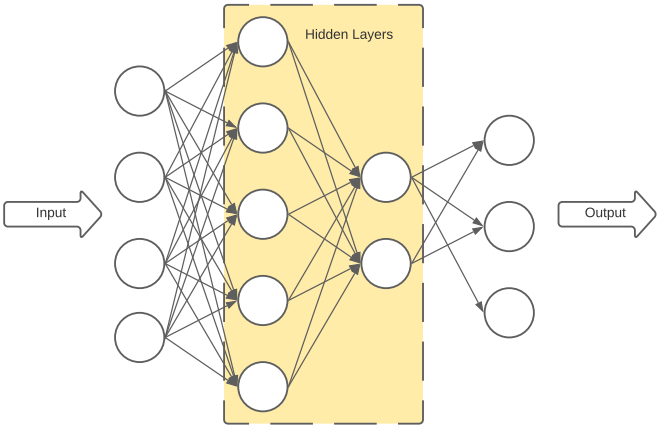


Fig. 1. The structure of a neural network. The circles represent neurons and each weighted connection is represented by an arrow. At each neuron, the weighted sum of the inputs is added to a bias term. The resulting sum is used as argument for the activation function to calculate the output of that neuron.

The aim of a neural network is to learn a function that maps the input to a desired output; a neural network is trained by supervised learning. The difference between the output of the network and the target output is quantified by a loss function. The purpose of the loss function is that it should punish the system when the output is different from the target. It is usually constructed so that it is at a minimum (or a maximum) when the target output is reached. The system produces different outputs by adjusting the weights of the connections and the bias terms of the neurons to compensate for each error found during learning. This is called backpropagation.

D. Generative Adversarial Network

A Generative Adversarial Network (GAN) is a machine learning algorithm designed to reproduce data following the

same distribution as some original data. It was first proposed in [13]. Generative Adversarial Networks have been used to reproduce several types of data, including human-like faces [14]. A GAN consists of two artificial neural networks which are called the *generator* and the *discriminator*. The networks compete against each other in a zero-sum game (i.e., one network's gain is the other's loss). The goal of the discriminator is to distinguish between real data and fake data generated by the generator. The goal of the generator is to fool the discriminator by generating fake data. The generator network takes Gaussian noise as its input and converts the noise into fake data through the network. The generated fake data is fed to the discriminator along with real data for classification. The classifications of real and fake data are compared to the true labels. The discriminator network learns to output classifications similar to the true labels. The generator learns to output data that the discriminator classifies as real. In Fig. 2 the training process of the GAN is visualized.

To exemplify the dynamic of the two networks, imagine that the discriminator is an arts professional and the generator is a forger. The forger makes a few counterfeits of famous paintings. The arts professional manages to reject the forged copies. However, the forger eventually learns what the professional looks for in real paintings and manages to fool the professional. The arts professional reacts by training more on real paintings to avoid future mistakes. Again, the forger's counterfeits eventually becomes even better. This repeats itself resulting in a constant tug of war strengthening the two networks as they learn from each other.

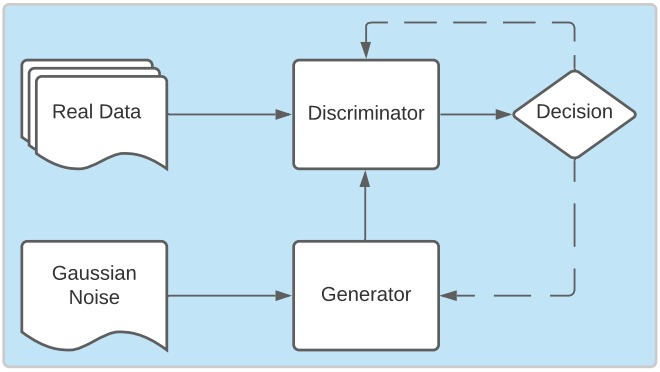


Fig. 2. The training process of a Generative Adversarial Network. The arrows pointed to the right represent the flow of data from their respective sources. The vertical arrow from the generator represents the generated samples. The arrow pointing to decision from the discriminator indicate that a decision has been made. The dashed arrows from the decision illustrate the back propagation process for the two networks.

III. Method

We go through our method leading up to the results of the project. In III-A, we introduce the dataset that we used to train the classifiers and the GAN. We also describe our pre-processing of the data and how we split it into training and testing sets. In III-B, we give a description of the CA system which was implemented in Python using the sci-kit learn library [15]. In III-C, we give a description of the implementation of the GAN based on Pytorch [16]. Lastly, the evaluation of the classifiers' performance is explained in III-D.

A. Dataset

1) Dataset description: In this project, we use data from the Touchalytics dataset [6] to train our classifiers and the GAN. The dataset contains stroke patterns of 41 users of mobile devices. The authors of [6] asked users to read three Wikipedia articles of various lengths on mobile devices. The users were free to read the documents in a pace of their choice and could navigate the documents freely by swiping with one or several fingers on the screen. For every stroke of the user's finger on the screen, the absolute time of the stroke, an event description (*finger up*, *finger down*, *finger move* or *multitouch*), as well as the phone orientation (*landscape* or *portrait*) was recorded. Also, for each finger touching the screen, the x- and y coordinates of the finger, the area of the finger, the pressure of the finger and the finger's orientation with respect to the screen was recorded. The authors proposed 30 features to define each stroke (e.g. *average velocity* and *stroke duration*).

2) Pre-processing: The raw feature set is available online [12]. We downloaded and pre-processed it to fit the need of this project. Firstly, we compared the strokes made during reading of the three separate documents. A classifier that was found to perform well on the strokes of each separate document was trained on strokes from a combination of all of the documents and we found that the classifier's performance did not decrease significantly. Thus, we deemed the strokes of the different documents similar and equally contributing to authenticating a user.

During closer examination of the dataset, we found that for some of the strokes, one or more feature values were missing (NaN). In other words, the information of these strokes were incomplete. Therefore, these entries were removed in the pre-processing phase. Moreover, the feature *inter-stroke time* that contained times between one stroke and the next, contained a handful of negative values, which suggests that these values were incorrect. These entries were also removed. The feature set also contained some abnormally large values. We found that by removing strokes with values deviating $k = 20$ or more standard deviations from the mean of the respective feature 8 consecutive times, we were able to remove abnormally large values without significantly changing the probability density of each feature, and without removing too many samples. By removing NaN entries, entries with negative *inter-stroke times* and entries with abnormally large values, the feature set was reduced from 11 888 to 11 500 samples. The lower limit of samples per user were set to 100 samples. This is based on the calculation of the metrics described in section III-D1. It was found that three users had a stroke number below the lower limit. They were excluded from the dataset.

Besides, some extra pre-processing was required for the categorical features present in the dataset, i.e., features with discrete values describing a state of the stroke, e.g., the feature *up/down/left/right flag* which had four states being *up*, *down*, *left* and *right* referring to the general direction of the stroke.

The values corresponding to the states were 1, 2, 3 and 4, respectively. A classifier trained on a categorical feature would take into account the numerical relationship of the values, in this case 2 being larger than 1, 3 being larger than 2, and so on. This is in fact an undesired effect, because swiping right is not inherently greater than swiping left as the numbers suggest.

To solve this problem, we introduced so called *dummy features* for every categorical feature. A dummy feature corresponds to a single state and the feature value of a stroke is valued 1 if the state is active and 0 if the state is inactive. In addition to *up/down/left/right flag*, the categorical features were *phone orientation* representing the way the mobile device was held at the time of the stroke, and *mid-stroke finger orientation* representing the finger orientation in the middle of the stroke. Combined across all three categorical features, nine states existed meaning the categorical features were replaced by nine dummy features (one for each state). This resulted in an increase in the amount of samples for every stroke from 30 to 36. As a final step in pre-processing the feature set, we standardized the set by subtracting the mean from every value and dividing with the standard deviation, feature-wise.

3) Splitting into training / testing datasets: The number of samples per user varied between 110 and 841 samples. For every user, the user samples were separated from all other samples into a user dataset $\mathbf{D_{i+}}$. From the user dataset, we randomly selected samples amounting to half the total samples for a user training set $\mathbf{D^t_{i+}}$. The remaining samples from the user dataset were extracted into a user testing set $\mathbf{D^s_{i+}}$. From the complementary subset $\mathbf{D_{i-}}$ (i.e., the subset containing samples from all other users), an equal amount of randomly selected samples per user were extracted into an other-user training set $\mathbf{D^t_{i-}}$, so that the other-user training set contained the same, or roughly the same, amount of samples as the user training set. In the same manner, we extracted randomly selected samples from the complementary subset into an other-user testing set $\mathbf{D^s_{i-}}$, so that it contained the same amount of samples as the user testing set. We then constructed the training set $\mathbf{D^t_i} = \mathbf{D^t_{i+}} \cup \mathbf{D^t_{i-}}$. Furthermore, we also created the testing set $\mathbf{D^s_i} = \mathbf{D^s_{i+}} \cup \mathbf{D^s_{i-}}$.

B. Continuous authentication system

We conceived a simple CA system made up of five state-of-the-art classifiers to serve as the subject of our evasion attacks. For the classifiers, three supervised classifiers and two unsupervised classifiers with implementations in Python [15] were chosen:

- Supervised classifiers:
  - Support Vector Machine (SVM),
  - Random Forest, and
  - Artificial Neural Network (ANN).
- Unsupervised classifiers:
  - One-class Support Vector Machine (One-class SVM), and
  - Isolation Forest.

For every user, the supervised classifiers were trained on the training set containing samples from the user (i.e., legitimate user samples) and samples from other users (i.e., illegitimate user samples), $\mathbf{D^t_i}$. The unsupervised classifiers were trained on the training set containing only legitimate user samples, $\mathbf{D^t_{i+}}$.

The training process of each classifier was controlled by a set of parameters. These are usually called hyper-parameters and will be referred to as such, henceforth. We did not consider changing all of the existing hyper-parameters of each model during training but we will mention here the hyper-parameters that were considered during this process. The behavior of the Support Vector Machine and the One-class Support Vector Machine are influenced by the kernel type $\in \{\text{linear, polynomial, radial basis function, sigmoid}\}$. Moreover, the kernel function could be further fine-tuned with a kernel coefficient $\gamma \in \mathbb{R}$ if the kernel type is not linear. There were two presets for the kernel coefficient: *auto* that is equal to $\frac{1}{n_{features}}$, where $n_{features} = 36$ is the amount of features of a sample, and *scale* that is a function of the variance in the dataset. For the Support Vector Machine, the strength of the regularization was controlled by a regularization parameter C which could take on positive values. A smaller value for C will ensure that the decision function of the classifier will return correct classifications on the training strokes while a larger value for C will reward a simpler decision function, even at the loss of classification performance on the training set, and hence decreasing the risk of overfitting [15]. Overfitting is when the decision function is made unnecessarily complex and provides good classifications on the training set but not on novel samples.

The Random Forest and Isolation Forest algorithms are *ensemble* algorithms that make classifications based on decisions of multiple base models. The number of models in each of these algorithms were considered hyper-parameters: the number of models in Random Forest n^R_{models} and the number of models in Isolation Forest n^I_{models}. The amount of features considered by every model in these algorithms to perform classification of a sample were also considered hyper-parameters: the number of features considered by every model in Random Forest $n^R_{features} \in \{n_{features}, \sqrt{n_{features}}, \log_2(n_{features})\}$ and the amount of features considered by every model in Isolation Forest $n^R_{features}$ which could be any number of available features per sample with an automatic setting of all features, $n_{features}$. Furthermore, the behavior of Random Forest was influenced by the criterion type $\in \{\text{Gini, entropy}\}$ which affects how each base model in the algorithm splits the samples into subsets that share attributes (i.e., performs classification). For the Isolation Forest, we could not control the rule for splitting.

The training process of the Artificial Neural Network depended on several hyper-parameters:

- Hidden layer sizes $n_{layers} = (n_1, n_2, ..., n_N)$, where the ith layer has n_i neurons, $i = 1, 2, ..., N$ and N is the number of hidden layers. The total amount of layers in the network is $N + 2$.
- Activation function $A \in \{\text{identity, logistic sigmoid, hyperbolic tan, rectified linear unit}\}$, where *identity* is equal to multiplying with 1, the *logistic sigmoid* function returns $A(s) = \frac{1}{1+e^{-s}}$, the *hyperbolic tan* function returns $A(s) = \tanh(s)$, the *rectified linear unit* function returns

$A(s) = \max(0, s)$ and s is the argument in equation (1).
- Optimizer $\in \{$limited-memory bfgs, stochastic gradient descent, Adam$\}$. The Adam optimizer is more suitable for larger datasets with more than 1000 samples while limited-memory bfgs could converge faster for smaller datasets [15].
- Regularization parameter α which is comparable to the regularization term C for the Support Vector Machine.

C. Generative Adversarial Network

To perform our evasion attacks, we implemented a Generative Adversarial Network to mimic the strokes of legitimate users. We used an implementation in Python based on Pytorch [16] which is licensed under the *MIT License*, see Appendix A. The configuration of the GAN were as follows. The discriminator and generator networks of the GAN had one hidden layer each and we were able to configure the number of neurons in most layers of the networks: the amount of neurons in the input layer of the generator network n_{gen}^{in}, the amount of neurons in the hidden layer of the generator network n_{gen}^{h}, and the amount of neurons in the hidden layer of the discriminator network n_{disc}^{h}. In addition to the layer sizes, the hyper-parameters of the GAN were the learning rate l_r and the number of epochs the GAN trained for N_e. We used binary cross entropy to calculate the networks' losses and the Adam optimizer to minimize the losses. The binary cross entropy $L(y, z)$ between the classification of the discriminator y and the correct label z is defined as

$$L(y, z) = -w \cdot [z \cdot \log(y) + (1 - z) \cdot \log(1 - y)], \qquad (2)$$

where w is a weight which was set to 1.

For every user, we trained the GAN to reproduce samples from the training dataset $\mathbf{D_{i+}^{t}}$. We labeled real samples from this dataset as 1 and samples generated by the generator as 0. The discriminator was trained to output the correct labels for real samples and generated samples by minimizing the binary cross entropy between the output of the discriminator and the correct labels (i.e., 1 and 0, respectively). On the other hand, the generator was trained to fool the discriminator by minimizing the binary cross entropy between the output of the discriminator on generated samples and the incorrect label (i.e., 1). After the training process, we generated a number of strokes equal to the amount of illegitimate user strokes in the training dataset $\mathbf{D_{i-}^{t}}$ into a generated user strokes dataset $\mathbf{D_{i*}}$. Lastly, we created a second testing set $\mathbf{D_{i}^{u}} = \mathbf{D_{i*}} \cup \mathbf{D_{i+}^{s}}$.

D. Evaluation

1) Metrics: To evaluate the quality of a classification process, the False Acceptance Rate (FAR) and False Rejection Rate (FRR) are often used. The FAR is defined as the percentage of authentication instances in which unauthorized users are falsely accepted and the FRR is the percentage of authentication instances in which authorized users are falsely rejected. The FAR and the FRR are functions of the classifier setting. A strict classifier will be much more inclined to reject users and therefore be more resistant to unauthorized users but be more obtrusive for authorized users. For a certain setting or *threshold* of the classifier, the classifier will return a pair of FAR and FRR values. The number of possible settings resulting in different FAR and FRR values depend on the amount of samples in the set. The lower bound of samples per user will provide a lower bound for the precision of FAR and FRR values. In our case, because the training set of each user contained at least 50 samples we obtained a precision of 2%.

In this project, we used the Equal Error Rate (EER) which combines the information from all FAR and FRR pairs; it is defined as the intersection of the FAR and FRR. We also defined the Threshold False Acceptance Rate (TFAR) as the FAR at FRR $= 10\%$. We conceived a scenario where a company may ask for a classifier with a FRR not exceeding 10% which means the classifier cannot reject more than 1 authorized user per 10 authentication instances of authorized users. The EER was approximated using a linear approximation between the two pairs of FAR and FRR values lying closest to the line $FAR = FRR$ on either side. As previously mentioned, the TFAR is the FAR at a specified FRR. Because the values of the FAR and the FRR were discrete, the FAR corresponding to the highest value of FRR less than 10% was chosen as the TFAR. See Fig. III-D1 for the approximation of the EER and the calculation of the TFAR.

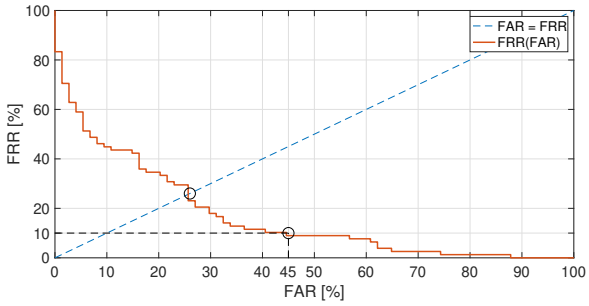


Fig. 3. False Rejection Rate plotted against False Acceptance Rate for example values. The circled intersection to the left in the figure represents the EER and the circled intersection to the right in the figure represents the TFAR.

2) Fine-tuning: To fine-tune the classifiers, and furthermore the GAN, we performed grid searches over the hyper-parameter spaces of each model (i.e., we trained and tested the models for different configurations of hyper-parameters). Fig. 4 show a visualization of the fine-tuning process. For every user, the classifiers were tested on the testing dataset $\mathbf{D_{i}^{s}}$ (i.e., some novel legitimate samples and some novel illegitimate samples) which is what we have referred to as *random evasion attacks*. In additions to this, the classifiers were tested on the second testing dataset $\mathbf{D_{i}^{u}}$ containing generated samples from the GAN which represents a sophisticated evasion attack. The classifiers were optimized with respect to the lowest average EER over users during random evasion attacks and the GAN was optimized indirectly to result in the highest average EER over users during sophisticated evasion attacks. During fine-tuning of the GAN, the classifiers with lowest average EER during random evasion attacks were employed.

Lastly, as a final run, the best GAN was used to attack

the best classifiers. This was repeated three times for every user (i.e., 3 times 38 users) which resulted in a total of 114 calculations of the metrics. The results were compared to the same metrics calculated during random testing of the best classifiers for the same repetitions.

IV. Results

A. Optimal hyper-parameters of the classifiers

We found that fine-tuning the classifiers using the proposed method was successful in decreasing the average EER over all users. In general, the default values of most parameters were found to be effective. The same settings were used for all users. Below are the values of hyper-parameters that were found to provide the best performance.

- Support Vector Machine:
  - Regularization parameter $C = 10$,
  - Kernel type: radial basis function,
  - Kernel coefficient $\gamma =$ auto
- Random Forest
  - Number of models $n^R_{models} = 100$
  - Amount of features $n^R_{features} = \log_2(36)$
  - Criterion type: gini
- Artificial Neural Network
  - Hidden layer sizes $n_{layers} = 100$
  - Activation function: logistic sigmoid
  - Optimizer: limited-memory bfgs
  - Regularization parameter $\alpha = 0.1$
- One-Class Support Vector Machine
  - Kernel type: linear
  - Kernel coefficient $\gamma =$ auto
- Isolation Forest
  - Number of models $n^I_{models} = 200$
  - Number of features $n^I_{features} =$ auto

The classifiers performed differently on different users which is expected seeing as the amount of samples per user varied. Another potential reason for this is the fact that some individuals are more chaotic in their behavior than others. In Table I, the average scores over users during random evasion attacks are presented for each classifier. Moreover, Fig. 5 and Fig. 6 show box plots demonstrating the distribution of the achieved EER and TFAR scores over the users, respectively.

TABLE I
Random evasion attacks Equal Error Rates and Threshold False Acceptance Rates

Classifier	Average EER	Average TFAR
One-class SVM	20 %	38 %
Isolation Forest	28 %	68 %
SVM	14 %	18 %
Random Forest	10 %	10 %
ANN	12 %	16 %

We see in Table I that the Random Forest classifier had best overall performance. Also, in general, the supervised classifiers outperformed the unsupervised classifiers with respect to both metrics during random evasion attacks. This was expected because the supervised classifiers are trained on some legitimate user samples, as well as some illegitimate user samples, while the unsupervised classifiers are only trained on legitimate user samples.

B. Optimal hyper-parameters of the Generative Adversarial Network

Regarding the optimization of the GAN, it was observed that the loss functions of the GAN did not converge to their optimal values for any configuration of the GAN, even for large numbers of epochs N_e. In all runs, the discriminator loss function decreased while the generator loss function increased, meaning the hyper-parameters that we tried resulted in the discriminator network overpowering the generator network during training. Nevertheless, the output of the GAN showed some similarity to real user strokes. For some features (and for some users), the reproduced feature values were similar to that of real strokes. For other features, the reproduced feature values were nowhere close real feature values. As an example, the probability density function of the generated feature values in the *inter-stroke time* feature for one of the users is shown in Fig. 7. In this case, the GAN produces a density similar to the original density. In contrast, we plotted an example where the probability density function of the generated feature values differed greatly from the training data in Fig. 8.

Regardless, the GAN was found to produce samples that resulted in higher average EERs over users for all the classifiers than during random evasion attacks; this was true for several configurations of the GAN. The combination of hyper-parameters and layer sizes that was found to produce the highest average EER over users on a single run were:

- learning rate $l_r = 3 \cdot 10^{-4}$,
- number of epochs $N_e = 100$,
- the amount of neurons in the input layer of the generator network $n^{in}_{gen} = 36$,
- the amount of neurons in the hidden layer of the generator network $n^h_{gen} = 216$,
- the amount of neurons in the hidden layer of the discriminator network $n^h_{disc} = 216$.

In Table II, we present the average EER and TFAR scores over users for generated evasion attacks. See also Fig. 9 and Fig. 10 that show box plots demonstrating the distribution of the achieved EER and TFAR scores over the users, respectively. Moreover, Fig. 11 visualize the increase in the metrics during sophisticated evasion attacks compared to random evasion attacks.

TABLE II
Generated attacks Equal Error Rates and Threshold False Acceptance Rates

Classifier	Average EER	Average TFAR
One-class SVM	22 %	62 %
Isolation Forest	38 %	90 %
SVM	38 %	76 %
Random Forest	24 %	54 %
ANN	38 %	76 %

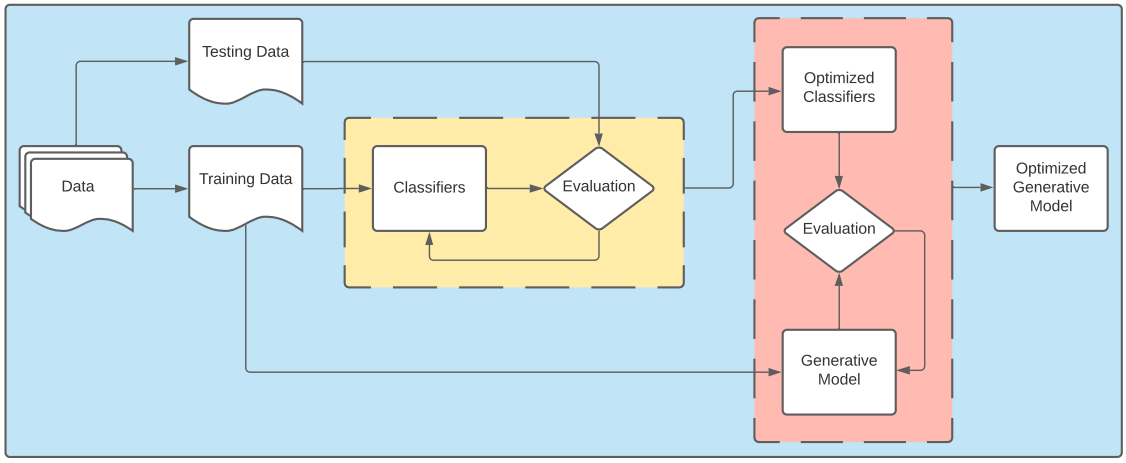


Fig. 4. Visualization of the fine-tuning process. The first dashed frame (from the left) shows the fine-tuning of the classifiers which results in the best configurations for the classifiers. The second dashed frame (from the left) shows the fine-tuning of the generative model using the best classifiers.

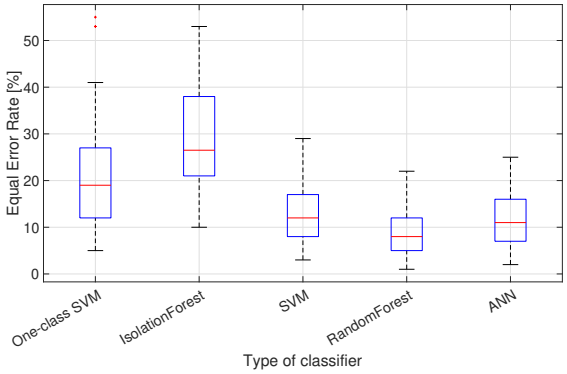


Fig. 5. Box plot of the achieved Equal Error Rates of the classifiers during random evasion attacks. In every box, the middle mark is the median Equal Error Rate, and the top and bottom edges are the 75th and 25th percentiles, respectively. The whiskers confine all values not considered outliers which are Equal Error Rates approximately 2.7 standard deviations from the mean. Outliers are seen as crosses outside of the whiskers.

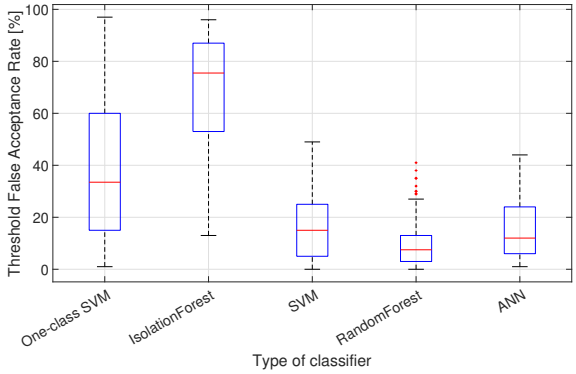


Fig. 6. Interpret as Fig. 5 but with Threshold False Acceptance Rate on the y-axis.

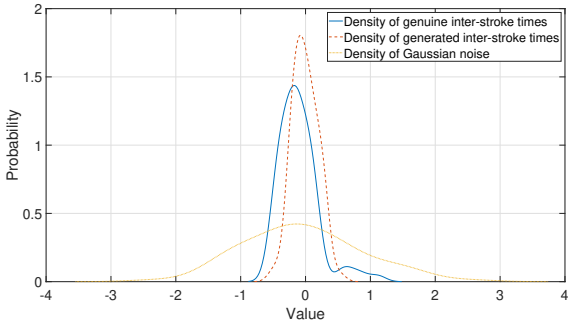


Fig. 7. Probability density estimate of genuine inter-stroke times, generated inter-stroke times and Gaussian noise with probability on the y-axis and value on the x-axis.

Most surprisingly, we found that the One-class Support Vector Machine was the best type of classifier against generated attacks (i.e., it achieved the lowest average EER and TFAR scores out of the classifiers). When it comes to the consistency of the classifiers, we found that the TFAR values over users for all classifiers except Isolation Forest showed a greater variance than during random evasion attacks. Moreover, we saw a larger decrease in the performance of the supervised classifiers than in the performance of the unsupervised classifiers. We reason that the unsupervised classifiers were more robust (i.e., showed the least decrease in performance against generated evasion attacks) because of their training method. A random sample that does not conform to any of the two classes known by the supervised classifiers, would still have to be classified as one of the classes and might therefore be classified as a legitimate user stroke about 50 % of the time. On the other hand, such a sample would be effortlessly rejected by the unsupervised classifier as an outlier (i.e., an illegitimate sample) because it doesn't match the learned characteristics of a legitimate user sample. The least robust classifier was the ANN whose EER and TFAR scores increased 26% and 60%, respectively.

V. Discussion

A. Optimality of results

Hyper parameter-tuning is a task of utmost importance in machine learning, which usually requires a lot of time and

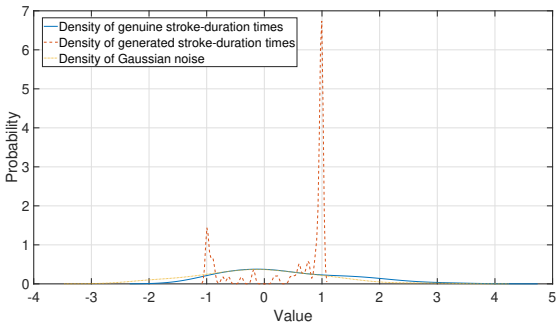


Fig. 8. Probability density estimate of genuine stroke-duration times, generated stroke-duration times and Gaussian noise with probability on the y-axis and value on the x-axis.

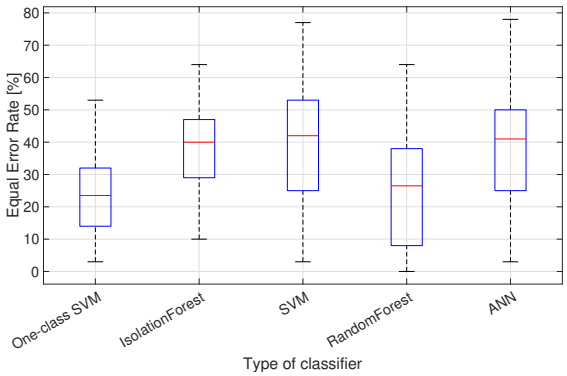


Fig. 9. Box plot of the achieved Equal Error Rates of the classifiers during a generated attack. Interpret as Fig 5.

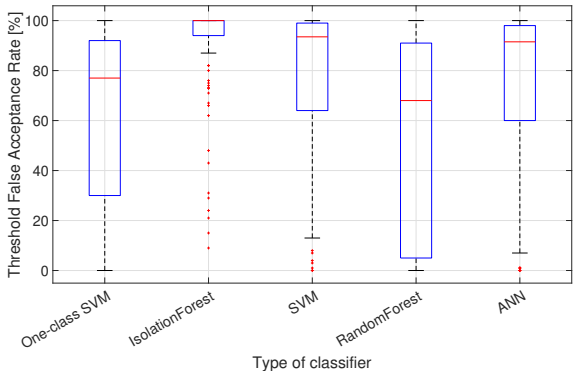


Fig. 10. Box plot of the achieved Threshold False Acceptance Rates of the classifiers during a generated attack. Interpret as Fig 6.

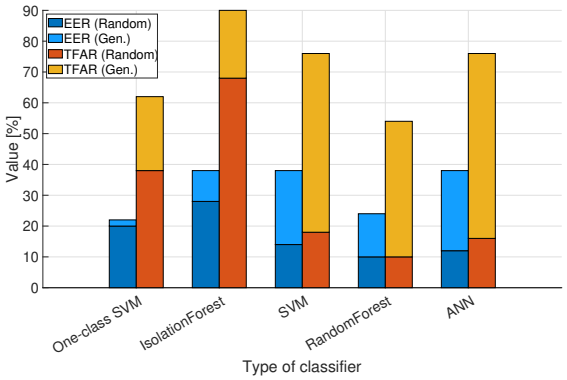


Fig. 11. Bar chart of the achieved average Equal Error Rates over users and the achieved average Threshold False Acceptance Rates over users. The increase in the metrics during sophisticated evasion attacks compared to random evasion attacks are stacked on top of the metrics during random evasion attacks.

computational power, not least for deep models like neural networks. To this end, we employed traditional grid searches which have some flaws compared to more sophisticated methods. The number of combinations of hyper-parameters tested for the classifiers and the GAN were limited based on the available computational power. As a result, even though the performance of the classifiers and the GAN are seen as satisfactory, it is unlikely that it is optimal. Further fine-tuning of the hyper-parameters could well lead to better performance, although it would be more computationally heavy.

B. Future work

One potential future direction for this project is to investigate the effect of imperfect attacker knowledge by, for example, removing features from the data or adding noise to the data. The idea is that removing data could mimic a real life scenario where a hacker acquires only a portion of the data e.g., by filming a user's swipe patterns and receiving features such as stroke speed and $x-$ and y coordinates, but missing a feature such as the pressure exerted on the screen. It would be interesting to see how well a GAN employed by the hacker would perform when data is missing. This extension to our project would also reflect the hacker's lack of insight regarding what features are used for verification.

On the same note, the method used to obtain user information might add some noise to the acquired data. Consider a hacker that is interested in obtaining spoken samples from an authenticated user to access a voice-recognition protected device. If the hacker records the authenticated user's voice using a microphone, it could have a negative effect on the sound quality. Therefore, we find it interesting to apply some noise to a dataset before training the GAN, in order to analyze the GAN's sensitivity to noise.

Another aspect to explore would be to compare different data types. One might ask themselves whether there exists some biometric data that is simpler for a GAN to mimic. In the case where a continuous authentication scheme is based on such biometric data, it could possibly be made more resistant to generated evasion attacks by using several types of biometric data.

As a final note, we would like to highlight conclusions drawn by the authors of [9]. They mention that in order to continue research in the field of continuous authentication, user privacy should not be overlooked. For example, the authors mention Support Vector Machines as non-privacy preserving classification algorithms because they store user profiles which lead to increased vulnerability. We suggest that further development of CA should be attempted to only consist of privacy-preserving algorithms.

VI. Summary

Continuous Authentication systems are based on individual biometric data. With recent advancements in the field of generative modeling, the security of these systems should be put under scrutiny. To this end, we investigated the security of CA on a publicly available dataset of stroke dynamics on mobile devices. First, we trained state-of-the-art classifiers. Next, we trained a Generative Adversarial Network on the same dataset to reproduce samples following the same distribution. Our results show that the performance of the classifiers significantly decreased on to the generated samples. Furthermore, we discussed the quality of our results and suggested future work in the field of continuous authentication related to generative models.

Appendix A
MIT license

Appendix B
Project colab notebook

Acknowledgment

The authors would like to address special thanks to the supervisor of the project, Ezzeldin Zaki, for lending his knowledge and support. Erik also wants to thank his girlfriend, Ella, for her patience during those long nights. Herman wants to thank his closest friend, God.

References

[1] Security Research Labs. (2021, Apr.) Fingerprints are not fit for secure device unlocking. [Online]. Available: https://srlabs.de/bites/spoofing-fingerprints/

[2] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge attacks on smartphone touch screens," in *Proceedings of the 4th USENIX Conference on Offensive Technologies*. USA: USENIX Association, Aug 2010, p. 1–7.

[3] Data Genetics. (2021, Apr.) Pin analysis. [Online]. Available: http://www.datagenetics.com/blog/september32012/

[4] N. Zheng, A. Paloski, and H. Wang, "An efficient user verification system using angle-based mouse movement biometrics," *ACM Trans. Inf. Syst. Secur.*, vol. 18, p. 27, Apr 2016.

[5] L. Lu, J. Yu, Y. Chen, H. Liu, Y. Zhu, L. Kong, and M. Li, "Lip reading-based user authentication through acoustic sensing on smartphones," *IEEE/ACM Transactions on Networking*, vol. 27, pp. 447–460, Feb 2019.

[6] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication," *IEEE Transactions on Information Forensics and Security*, vol. 8, pp. 136–148, Jan 2013.

[7] T. Feng, Z. Liu, K.-A. Kwon, W. Shi, B. Carbunar, Y. Jiang, and N. Nguyen, "Continuous mobile authentication using touchscreen gestures," in *2012 IEEE Conference on Technologies for Homeland Security (HST)*, Nov 2012, pp. 451–456.

[8] Z. Sitová, J. Šeděnka, Q. Yang, G. Peng, G. Zhou, P. Gasti, and K. S. Balagani, "Hmog: New behavioral biometric features for continuous authentication of smartphone users," *IEEE Transactions on Information Forensics and Security*, vol. 11, pp. 877–892, May 2016.

[9] M. Al-Rubaie and J. M. Chang, "Reconstruction attacks against mobile-based continuous authentication systems in the cloud," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2648–2663, Dec 2016.

[10] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, p. 15, Sep 2016.

[11] P. Bontrager, A. Roy, J. Togelius, N. Memon, and A. Ross, "Deepmasterprints: Generating masterprints for dictionary attacks via latent variable evolution*," in *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, Apr 2018, pp. 1–9.

[12] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song. (2021, Apr.) Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. [Online]. Available: http://www.mariofrank.net/touchalytics/index.html

[13] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, Dec 2014, p. 2672–2680.

[14] T Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen and T. Aila. (2021, Apr.) This person does not exist. [Online]. Available: https://thispersondoesnotexist.com/

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[16] A. Persson. (2021, Apr.) Machine-learning-collection/ml/pytorch/gans/1. simplegan/. The code is licensed under the MIT license and a complementary video explaining the code exists at YouTube titled Building our first simple GAN. [Online]. Available: https://github.com/aladdinpersson/Machine-Learning-Collection/blob/master/ML/Pytorch/GANs/1.\%20SimpleGAN/fc_gan.py

CONTEXT M – PART I
INFORMATION ENGINEERING: BIG DATA & AI

POPULAR DESCRIPTION

AI - a dangerous beast or a useful tool?

Humans are considered to be the most dangerous animal on the planet. Imagine what horrifying creations we could produce. We are obviously talking about Artificial Intelligence, AI. Could it be caged? Will we be able to tame our own creations?

When talking about AI, a common fear people have is that it will eventually outsmart us. When in fact, it already has. Try playing a game of chess against Google's AlphaZero. You *will* lose, and you will lose because AlphaZero is optimized and trained specifically for this task - a property that is highly desirable and advantageous when applied to other specific assignments.

A famous rule is the *10 000 hours rule*, which states that in order to master any talent you have to practise it for 10 000 hours. Meanwhile within 24 hours AlphaZero had achieved a superhuman level of chess. If AI is capable of performing tasks that much faster than a regular human being, the possibilities are endless!

Mastering the art of chess merely scratches the surface of optimization applications in which we could utilize AI. Given large data sets, AI can extract the most important features and draw relevant conclusions in order to perform its given task. For example, AI can lead the way in next generation DNA sequencing. By using AI to process a signal from an infinitely long single stranded DNA, it can accurately predict the genetic sequence. Another application within medicine is image processing in which AIs have been shown to perform just as well as human experts, when making medical diagnoses. It is only a matter of time before AI learns to be much better, freeing up more resources within healthcare to better help patients.

As the amount of collected data increases in our lives, AI becomes the natural choice for a companion in managing the data. AI will not turn into the wild animal we fear, but rather help us optimize our way of living and pave the way to societal efficiency.

SUMMARY OF PROJECT RESULTS

Big Data and AI are two fields that go hand in hand. In order to process the large amounts of data collected in the modern digital world, it is necessary to implement reliable and intelligent solutions. The following projects contribute, in various ways, to the fields by optimizing processes related to the extraction and interpretation of information from large datasets.

The M1 project group investigated different methods for variable selection in high-dimensional data, and studied to which extent the subset of selected variables differ depending on which method is used. Furthermore, the project group evaluated how combining the result of these methods could lead to improving the choice of important variables. Amongst the variable selection methods used in this project were Lasso, Elastic net and Ridge Regression. Implementing the methods on different kinds of datasets leads to the conclusion that the methods give different subsets of selected variables, both regarding the size of the subset and which variables are selected in the subset. Therefore, combining the results of different methods gives a more reliable final model. This result can be used for more accurately selecting important features of large data sets of any kind, as well as predicting future values.

In future studies in this field, it could be of interest to implement the different methods mentioned above on several different kinds of dataset, for example data sets with grouped or missing variables. Another improvement in this field could be achieved by creating a concrete model for how to combine the results of the variable selection methods.

The M2 project group mainly analyzed compression and distribution of neural networks with possible IoT applications. State of the art methods (pruning and knowledge distillation) were used in order to train smaller student models, optimized for IoT devices, to mimic a larger model. The intention of such models is to enable decentralized decision making, utilizing parallel computing. In addition, the methods also serve to minimize communication between devices, thereby increasing computing efficiency. Commonly, a central processing unit is used in IoT networks in order to process collected data, which raises concerns about data breaches. Distributing a model on several devices could reduce the risk of such security concerns.

Further improvements of the methods could include applying similar models to a multi-classification problem and seeking alternative ways of distributing the model. This could result in a higher hedge against data breaches by assuring the input data is not shared by the edge devices. Additionally, an investigation of the improvements in computational times emerging from distributing the model could be considered. The present study merely considers the size and accuracy of the models.

The M3 project group investigated different attributes of electrical signals generated by Oxford Nanopore DNA sequencing. Nanopore sequencing is one of the latest generation sequencing technologies in which a DNA or RNA molecule is fed through a nanopore (a nano-scale hole). This causes changes in an ionic current passed through the nanopore, and results in a unique electrical signal that can identify the nucleotide bases of the molecule. The aim of the project was to determine if it is possible to model and parameterize signal data from a DNA strand, in order to segment it in its nucleotide bases. In this project, methods were developed to analyze the significance of various features extracted from the signal data, such as the sudden changes of a signal in relation to time, or the number of zero crossings of a signal, and their correlation to the probability distribution that best describes the signal.

The results of the project will help Nanopore sequencing developers in the later stages of assigning nucleotide bases to the electrical current signal. Once refined and improved, Nanopore sequencing will enable portable, and cheaper sequencing compared to current sequencing methods. In future projects within Nanopore DNA sequencing, more complex distributions could be investigated to analyze whether they yield more accurate results when implemented with an estimator. Further research can also be done within neural networks and their effectiveness in data segmentation, as well as more advanced methods for distinguishing homopolymers.

IMPACT ON SOCIETY AND ENVIRONMENT

The definition of Big Data and Artificial Intelligence is perhaps to many somewhat vague or unfamiliar. More clear is probably the purpose - to extract valuable information and enable rational and optimized decision making processes.

However, the field Big Data and AI also presents issues that are essential to address and discuss. Knowledge comes with responsibility, and the natural question to ask when any process is automated is - Who will be held accountable if the outcome of the process is undesired? Undoubtedly, this raises juridical challenges, which will have to be agreed upon before deployment of such automation.

Implementation of AI on Big Data can lead to more objective and rational decision making processes by minimizing human errors and biases. Considering the complexity of modern data sets, drawing any conclusions is sometimes impossible for the human brain - a task an AI could do with ease. Automated choices could, for instance, be applied to Smart Home networks, which are then able to adjust and direct household energy consumption, thereby reducing both the consumer’s expenses while also lowering the environmental impact.

Nonetheless, such a mechanism carries with it the drawback of collecting personal data, which is one of the main issues regarding Big Data. When data contains personal or identifying information, it is especially important that it is securely stored and that it is acquired with the affected parties' consent. Another important issue is how the data is used and presented so that unintentional data leaks don’t occur. An example of this is the case of DNA sequencing. Cheaper sequencing methods could potentially lead to large scale research projects within medicine and social studies. However, sharing data on such a large scale could also lead to cases of misuse or exploitation of people’s personal data. DNA reads contain the entire genetic makeup of an individual, and it is therefore also a possibility that the reads are used for purposes other than the ones explicitly expressed. This requires that clear guidelines are in place so that consumers, and any other data sources, are properly

informed about the different ways their personal data could be used. With proper laws and regulations, people's privacy and security can be protected.

Furthermore, intelligent decision making is a way of reducing subjectivity in determination in important matters such as politics, health issues and juridical assessments. Because of the objectivity of AI, decisions are solely based on collected data. This makes the decision making, for example in recruitment, unbiased which eliminates the risk of personal prejudices to affect the process. An important aspect is that the data itself needs to be unbiased.

To summarize the preceding arguments, we believe that it is of major interest to use Big Data and AI in order to create reliable and objective models. The results can be applied in interminably different areas, like health and environment, and lead to more efficient systems and reliable conclusions. An important aspect when using Big Data is that the data is handled properly, since with Big Data comes big responsibility!

Variable Selection in High-Dimensional Data

Johan Hallberg and Sarah Reichhuber

***Abstract*—Estimating the variables of importance in inferential modelling is of significant interest in many fields of science, engineering, biology, medicine, finance and marketing. However, variable selection in high-dimensional data, where the number of variables is relatively large compared to the observed data points, is a major challenge and requires more research in order to enhance reliability and accuracy. In this bachelor thesis project, several known methods of variable selection, namely *orthogonal matching pursuit* (OMP), *ridge regression*, *lasso*, *adaptive lasso*, *elastic net*, *adaptive elastic net* and *multivariate adaptive regression splines* (MARS) were implemented on a high-dimensional dataset. The aim of this bachelor thesis project was to analyze and compare these variable selection methods. Furthermore their performance on the same data set but extended, with the number of variables and observations being of similar size, were analyzed and compared as well. This was done by generating models for the different variable selection methods using built-in packages in R and coding in MATLAB. The models were then used to predict the observations, and these estimations were compared to the real observations. The performances of the different variable selection methods were analyzed utilizing different evaluation methods. It could be concluded that some of the variable selection methods provided more accurate models for the implemented high-dimensional data set than others. Elastic net, for example, was one of the methods that performed better. Additionally, the combination of final models could provide further insight in what variables that are crucial for the observations in the given data set, where, for example, variable 112 and 23 appeared to be of importance.**

***Sammanfattning*—Att skatta vilka variabler som är viktiga i inferentiell modellering är av stort intresse inom många forskningsområden, industrier, biologi, medicin, ekonomi och marknadsföring. Variabel-selektion i högdimensionella data, där antalet variabler är relativt stort jämfört med antalet observerade datapunkter, är emellertid en stor utmaning och kräver mer forskning för att öka trovärdigheten och noggrannheten i resultaten. I detta projekt implementerades ett flertal kända variabel-selektions-metoder, nämligen *orthogonal matching pursuit* (OMP), *ridge regression*, *lasso*, *elastic net*, *adaptive lasso*, *adaptive elastic net* och *multivariate adaptive regression splines* (MARS), på ett högdimensionellt data-set. Syftet med detta kandidat-examensarbete var att analysera och jämföra resultaten av dessa metoder. Vidare analyserades och jämfördes metodernas resultat på samma data-set, fast utökat, med antalet variabler och observationer ungefär lika stora. Detta gjordes genom att generera modeller för de olika variabel-selektions-metoderna via inbygga paket i *R* och programmering i MATLAB. Dessa modeller användes sedan för att prediktera observationer, och estimeringarna jämfördes därefter med de verkliga observationerna. Resultaten av de olika variabel-selektions-metoderna analyserades sedan med hjälp av ett flertal evaluerings-metoder. Det kunde fastställas att vissa av de implementerade variabel-selektions-metoderna gav mer relevanta modeller för datan än andra. Exempelvis var elastic net en av metoderna som presterade bättre. Dessutom drogs slutsatsen att kombinering av resultaten av de slutgiltiga modellerna kunde ge en djupare insikt i vilka variabler som är viktiga för observationerna, där, till exempel, variabel 112 och 23 tycktes ha betydelse.**

***Index Terms*—variable selection, variable selection methods, linear regression, high-dimensional data, variable importance.**

Supervisors: Prakash Borpatra Gohain and Magnus Jansson

TRITA number: TRITA-EECS-EX-2021:185

I. INTRODUCTION

The aim of this bachelor thesis project is to study the concept of variable selection and further investigate the difference between several kinds of variable selection methods. Comparing these methods can be done using various evaluation methods as for example root mean squared error (RMSE). Though it is of great interest to compare different variable selection methods, there is no easy way to do it. In studies of, for example, Zou and Zhang [1] and Hastie et al. [2], some of the variable selection methods used in this report were presented and compared. But there is no obvious rule for how to compare them and what variable selection method is the most appropriate for a certain data set. The main issue is that every high-dimensional data set is different from the other. Thereby, if for one data set it is obvious that one variable selection method is superior in performance, there is still no assurance that it will be the best performing one for another data set.

In this project, a data set containing 122 independent variables and 123 measurements will be used. The same data set was also adjusted so that the number of measurements decreased to 50. The aim is to find the differences in the final models of each variable selection methods for the two different sizes of the data set. Furthermore, the differences in the results between the variable selection methods is analyzed and discussed. It is also of interest to evaluate if combining results across different variable selection methods could lead to improved models. Therefore, a few innovative ideas will be presented about this topic.

Another study similar to this has been done by Lima et al. [3]. In that study, a number of known variable selection methods, some being the same as the ones used in this report, are analyzed for a high-dimensional data set. The aim was to evaluate the extent to which variable selection changes for different methods, and also whether combining results of the methods could enhance the data interpretation. The authors furthermore evaluated the stability of the selected independent variables of each method, from which they could derive what variables were of major importance. The aim of this project is thereby to further investigate the different variable selection methods but for another data set.

The remaining part of the thesis is organized as follows. The theory behind multivariate linear regression, as well as for each

variable selection method, will be presented in Section II. The implementation of each variable selection method in R and MATLAB will be described in Section III. In Section IV, the selected variables and model performances for each variable selection method are visualised. Comparison of the results of the different variable selection methods and discussion about future studies are made in Section V. The conclusion of this bachelor thesis project is presented in Section VI.

II. Background

In this Section, the background theory needed for the bachelor thesis project is presented in detail. First, the theory behind linear regression and variable selection is given. Then the relevant theory behind all the variable selection methods used in this project and cross validation will be mentioned. Lastly, model evaluation will be presented.

A. Linear Regression

Regression analysis is a technique for explaining the relationship between variables. Often we have a given set of data points x and y and we want to find a relationship between these variables. Assuming a linear relationship between them leads to the specific field of regression analysis called *linear regression*. Generally these kinds of models can be written on the form

$$\boldsymbol{y} = \beta_0 + \beta_1 \boldsymbol{x} + \boldsymbol{\varepsilon} \tag{1}$$

where $\boldsymbol{y}$ is the vector of observations, also referred to as the dependent variables. The $\boldsymbol{x}$-vector, known as the regressor or predictor, is the independent variable. $\boldsymbol{y}$ is also determined by the coefficient β_1 and the constant term β_0. $\boldsymbol{\varepsilon}$ represents the model error. The model with only β_1 and β_0 is called the simple linear regression model. This model can be visualised in Figure 1, where β_0 is the intercept and β_1 the slope of the curve fitted to the set of data points.

Multiple linear regression is comparable to simple linear regression except in the way that there is more than one independent variable in this case. These models can be described by the equation

$$\boldsymbol{y} = \beta_0 + \beta_1 \boldsymbol{x}_1 + \beta_2 \boldsymbol{x}_2 + ... + \beta_p \boldsymbol{x}_p + \boldsymbol{\varepsilon}. \tag{2}$$

Here each $\boldsymbol{x}_j$, for $j = 1, 2, ..., p$, corresponds to a predictor and β_j is a coefficient. p is the number of predictors. If all of

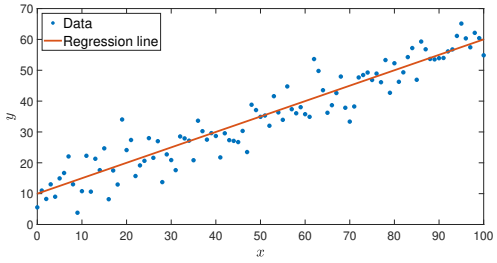


Fig. 1. An example of simple linear regression. Dots are data points and the line is a linear function fitting the data points called the *regression line*

the values of the predictors in the data set are standardised, this means that each column of the $\boldsymbol{X}$ matrix is centered as

$$\frac{1}{m}\sum_{i=1}^{m} x_{ij} = 0 \tag{3}$$

with unit variance according to

$$\frac{1}{m}\sum_{i=1}^{m} x_{ij}^2 = 1 \tag{4}$$

where i represents each row in X. m describes the number of measurements. If the observations also are centered as

$$\frac{1}{m}\sum_{i=1}^{m} y_i = 0. \tag{5}$$

the intercept term, β_0, in (2) can be neglected. More about standardisation of data can be read about in Hastie et al. [2].

There are m number of measurements in the data set, each depending on p different predictors. The column vectors $\boldsymbol{x}_j$ can be assembled into a matrix. Neglecting the intercept term β_0, leads to that (2) can be rewritten as

$$\mathbf{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{6}$$

where $\boldsymbol{X}$ is a matrix consisting of p number of columns and m number of rows, see Figure 2. Each row in $\boldsymbol{X}$ is a specific measurement that depends on the predictors, $\boldsymbol{x}_j$, in $\boldsymbol{X}$. The aim is to estimate $\boldsymbol{\beta}$ such that it minimize the least squares cost as

$$\arg\min_{\boldsymbol{\beta}} \left\{ ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2 \right\} \tag{7}$$

Here "$||\cdot||_2$" denotes the usual Euclidean norm of vectors. $\boldsymbol{\beta}$ is often estimated using the least squares solution given by

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}. \tag{8}$$

After finding appropriate β-coefficients for the given data set, a model with these coefficients can be formed as

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}}. \tag{9}$$

The goal is that the estimated observations, $\hat{\boldsymbol{y}}$, should be as close to the real observations, $\boldsymbol{y}$, as possible. The estimated β-coefficients can also be used for predicting observations $\boldsymbol{y}$ not yet evaluated, given new measurements. To ensure that the calculated $\hat{\boldsymbol{y}}$ is a good estimated observation, and thereby a good model, different evaluation methods can be used. These evaluation methods will be described further in Section II-E. More about linear regression can be read about in Montgomery et al. [4] and in Sridharan [5].

B. Variable Selection

The purpose of variable selection is to estimate what predictors that are of importance for the observations in a given data set. In data sets with the numbers of predictors p being larger than the number of measurements m, it is often the case that only a few of these predictors are actually crucial for the resulting observations $\boldsymbol{y}$. The predictors $\boldsymbol{x}_j$ that appear to have a considerable impact on the observation vector $\boldsymbol{y}$ are here referred to as *important* variables. As seen in Figure 2,

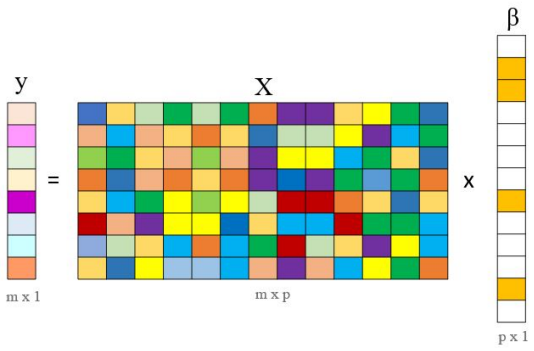


Fig. 2. A general system with m measurements and p predictors, where some β-coefficients are active and some are not. The colours represent the randomness of the data.

this results in that only some of the β-coefficients in the $\boldsymbol{\beta}$-vector are *active*, i.e not white. These β-coefficients relate to the predictors, namely the columns of the $\boldsymbol{X}$ matrix, that are important.

Having $m < p$ results in that the least squares solution often overfits the data, and does not perform optimal. Therefore, other methods for estimating the linear regression model, and thereby the β-coefficients, have been derived. Methods on how to estimate these coefficients, and thereby the important variables, are often referred to as *variable selection methods*. To find these coefficients a data set is used with given values of the regressors $\boldsymbol{X}$ and the observations $\boldsymbol{y}$. The variable selection methods evaluated in this bachelor thesis project are thoroughly described next, in Section II-C. More about linear regression and variable selection can be read about in Montgomery et al. [4] and in Sridharan [5].

C. Variable Selection Methods

In this report several known methods for variable selection will be studied - for the sake of simplicity they will from now on be referred to as *methods*. The methods differ from each other in several aspects, among other things in how they choose the set of important variables. The methods analyzed in this report are *OMP*, *ridge regression*, *lasso*, *elastic net*, *adaptive lasso*, *adaptive elastic net* and *MARS*. The theory behind respective method is presented below.

OMP, orthogonal matching pursuit, is a greedy iterative method for variable selection - greedy meaning that it at each step does the selection according to what gives an immediate benefit, not regarding future impacts. At each iteration $k = 1, 2, ..., p$ the OMP algorithm selects only one column of the $\boldsymbol{X}$ matrix and moves it to the set of selected independent variables, which will be called S. Thus, after k iterations, the set S will contain k selected variables. As stated in Cai and Wang [6], the OMP algorithm at each step selects the variable $\boldsymbol{x}_j$, for $i = 1, 2, ..., p$, most correlated to the current residual $\boldsymbol{r}_{k-1}$, meaning the variable that maximizes

$$|\langle \boldsymbol{x}_i, \boldsymbol{r}_{k-1} \rangle|. \tag{10}$$

The residual, for which the definition will be presented later, will be set to the observation vector before the first iteration, $\boldsymbol{r}_0 = \boldsymbol{y}$, and the correlations are calculated by the dot product of the residual and each predictor.

The next iteration, the OMP algorithm will select a new independent variable out of the ones that are left, according to which is the most correlated to the updated residual. It then adds that variable to the set S.

The residual to be used in the next iteration is calculated by using the least squares estimate of $\hat{\boldsymbol{\beta}}$, namely

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}_S^T \boldsymbol{X}_S)^{-1} \boldsymbol{X}_S^T \boldsymbol{y} \tag{11}$$

where $\boldsymbol{X}_S$ is the matrix in which each column is one of the selected variables in S and $\boldsymbol{y}$ is the given vector of observations. Using the $\hat{\boldsymbol{\beta}}$-vector from (11), the estimated value of $\boldsymbol{y}$ is given by (9), from which the current residual, $\boldsymbol{r}_{k-1}$, is derived by

$$\boldsymbol{r}_{k-1} = \boldsymbol{y} - \hat{\boldsymbol{y}}. \tag{12}$$

This leads to that after a predictor is selected, the explained portion of $\boldsymbol{y}$, namely $\hat{\boldsymbol{y}}$, is removed from the residual. Thereby the remaining residual will be orthogonal to the selected predictors, which prevents them from being selected again. If no stop condition is given, the OMP algorithm will eventually select all predictors. More about OMP can be read about in Cai and Wang [6].

Ridge regression is a penalizing method which means that it shrinks β-coefficients towards zero. Since it rarely shrinks the β-coefficients to exactly zero, it usually selects all variables. Ridge regression solves the optimization problem

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{ridge}} = \underset{\beta_0, \boldsymbol{\beta}}{\arg\min} \left\{ \frac{1}{2m} \sum_{i=1}^{m} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j)^2 \right\} \tag{13}$$

subject to

$$\sum_{j=1}^{p} \beta_j^2 \leq t^2. \tag{14}$$

(13) is the cost function that we are trying to minimize with respect to $\boldsymbol{\beta}$. (14) is the constraint on $\boldsymbol{\beta}$, where $t \geq 0$ is called the *bound* and limits the sum of the β-estimates. The constraint can also be characterised with the ℓ_2-norm constraint

$$||\beta||_2^2 \leq t^2. \tag{15}$$

The constraint region for ridge regression can be visualized as the blue area on the right hand side of Figure 3, here $p = 2$. The residual sum of squares (RSS), defined by

$$\text{RSS} = \sum_{i=1}^{m} (y_i - \hat{y}_i) \tag{16}$$

has elliptical curves centered at $\hat{\beta}$ which, in Figure 3, corresponds to the least squares solution where the RSS is minimized. The constraint region in Figure 3 can be expressed as

$$\beta_1^2 + \beta_2^2 \leq t^2. \tag{17}$$

The solution to (13) is found when the elliptical curve contacts the constraint region. For ridge to be able to shrink the β-coefficients to exactly zero, the elliptical curves has to tangent the constraint region where some β-coefficient is equal to zero.

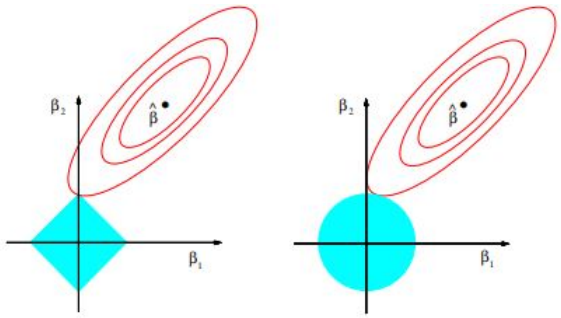


Fig. 3. For the model $y = \beta_1 x_1 + \beta_2 x_2$, the figure represents the constraints, i.e. the blue areas, of Lasso (left), defined by $|\beta_1| + |\beta_2| \leq t$ and Ridge Regression (right), defined by $|\beta_1^2| + |\beta_2^2| \leq t^2$. $\hat{\beta}$ is the least-squares estimate and the red ellipses show the residual-sum-of-squares function [2].

Generally, the predictors in $\boldsymbol{X}$ are standardized as described by (3) and (4). Furthermore, the observations y_i are often centered, see (5). This centering condition leads to that the intercept β_0 can be neglected. Without the intercept, (13) and (14) can be alternatively formulated using a *Lagrange multiplier* as

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{ridge}} = \arg\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2m} ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2 + \lambda ||\boldsymbol{\beta}||_2^2 \right\} \tag{18}$$

where $\lambda \geq 0$ is the Lagrange multiplier, also termed the regularization parameter. The last term in (18) is the ℓ_2-penalty. There is a distinct relationship between the constrained problem, (13) and (14), and the Lagrangian problem, (18). For each value of t in the range where the constraint, (14), is fulfilled, there is a corresponding value of λ giving an equivalent solution from the Lagrangian form, (18). The purpose of regulating the bound or the Lagrange multiplier, for (14) respectively (18), is to determine how strictly the penalization is going to be. For instance, a high value of t, respectively a small value of λ, allows numerous predictors to be selected. Meanwhile, a low value of t, respectively a high value of λ, leads to fewer predictors being selected. As mentioned earlier - all predictors are technically selected for ridge regression, but for other methods this can lead to a sparse model. For the remainder of this report, the Lagrangian form will be used when defining ridge regression.

Ridge regression is appropriate when the number of predictors is higher than the number of measurements, $p \gg m$. One considerable weakness with ridge regression is the fact that it selects all variables, resulting in a final model that has no sparsity. Sparsity is desired, since it clearly shows which variables that are important. Further reading about ridge regression can be done in Hastie et al. [2] and Ginestet [7].

Lasso is similar to ridge regression in several ways and will therefore be compared to ridge throughout the theory part. As ridge regression, lasso is a penalizing method but with the difference that it can actually shrink the β-coefficient to exactly zero. This leads to that lasso can generate a sparse model with only a few important variables being selected. In Lagrangian form, lasso estimates the β-coefficients as

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{lasso}} = \arg\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2m} ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2 + \lambda ||\boldsymbol{\beta}||_1 \right\} \tag{19}$$

where $\lambda \geq 0$. The disparity from ridge regression can easily be shown in the constraint for lasso, given by

$$\sum_{j=1}^{p} |\beta_j| \leq t \tag{20}$$

compared to the constraint for ridge regression, see (14). The constraint can be characterised with the ℓ_1-norm constraint

$$||\beta||_1 \leq t. \tag{21}$$

For the example presented for ridge, where $p = 2$, the difference in the constraints for lasso can be visualised on the left hand side of Figure 3. Because of the sharp edges in the lasso constraint, the elliptical curves will often tangent the edges where, for this example, either β_1 or β_2 is zero. The lasso constraint will always consist of sharp edges even when $p > 2$. For high values of λ, lasso gives a sparse model while for small values it selects more predictors.

The ℓ_1-penalties that lasso uses provide an understandable way to accomplish sparsity in the model. The fact that ℓ_1-penalties are convex and encourage sparsity may result in serious computational advantages. Lasso can at most select m number of predictors. If a data set consists of a large number of predictors and a few number of measurements, $p \gg m$, then at most m number of predictors will be non-zero in the model, which makes the computation much simpler. Though it makes the computation simpler it is on the other hand a limitation for lasso in some cases. Further reading about lasso can be done in Hastie et al. [2] and in Hastie [8].

Elastic net utilizes a combination of ℓ_1- and ℓ_2-penalty, and can therefore be described as a combination of lasso and ridge regression. By using the parameter $\alpha \in [0, 1]$, elastic net makes a compromise between lasso and ridge regression penalties, and the regression coefficient estimates are obtained by

$$\begin{aligned} \hat{\boldsymbol{\beta}}_{\boldsymbol{enet}} = \arg\min_{\boldsymbol{\beta}} \Big\{ & \frac{1}{2} ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2 + \\ & + \lambda \big[\tfrac{1}{2}(1-\alpha) ||\boldsymbol{\beta}||_2^2 + \alpha ||\boldsymbol{\beta}||_1 \big] \Big\}. \end{aligned} \tag{22}$$

Setting $\alpha = 1$ results in that (22) utilizes the ℓ_1-norm, or lasso penalty, as in (19). Similarly, setting $\alpha = 0$ results in the ℓ_2-norm and the ridge regression penalty as in (18). The terms $\frac{1}{2}$ and $\frac{1}{2m}$ in the mentioned equations are merely for mathematical reasons, and do not qualitatively affect the final result. A graphical comparison between ridge regression, lasso and elastic net penalty in the two-dimensional case is presented in Figure 4, where the elastic net penalty is given by

$$(1-\alpha) ||\boldsymbol{\beta}||_2^2 + \alpha ||\boldsymbol{\beta}||_1. \tag{23}$$

The advantage of elastic net, compared to ridge regression, is that it like lasso can set coefficients to zero. This is a good property since it is often desired to have a sparse

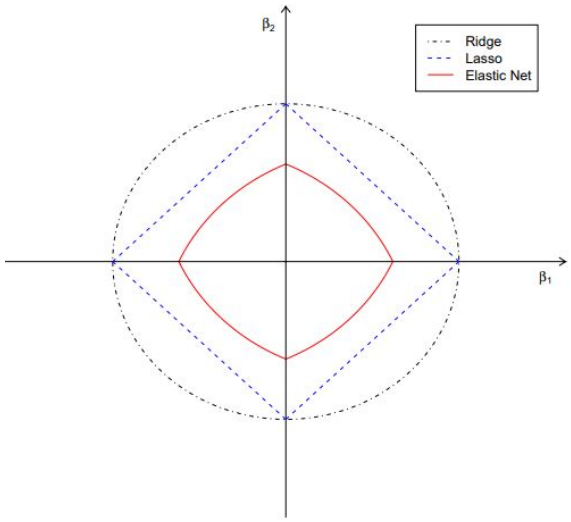


Fig. 4. For the model $y = \beta_1 x_1 + \beta_2 x_2$, the outermost circular contour represents the ridge constraint, the rectangular contour in the middle describes the lasso constraint and the innermost contour describes the combined constraints, namely the elastic net constraint [8].

model, both in a computational matter but also for extracting what variables of the data set that are the most important. Furthermore, elastic net often outperforms lasso when the number of predictors, p, is much larger than the number of observations m. It is also superior to lasso in the matter of selecting grouped variables, i.e. it lets strongly correlated independent variables be either all inside or all outside the final set of important variables. Lasso, however, often only selects one of the correlated variables as important. Elastic net also has the ability to select all p predictors, which lasso has not. In Figure 4, the sharp corners of the elastic net penalty represents the property of variable selection, as seen in lasso, while the curved contours represent the property of sharing coefficients, as in ridge regression. Both Hastie and Tibshirani [2] and the inventors of the method, Zou and Hastie [8], provide all details regarding the elastic net method.

Adaptive lasso is an extension of lasso, meaning it as well employs ℓ_1-penalty. The adaptive lasso method was first published by Zou [9]. In this method, adaptive weights are used to penalize each of the $\boldsymbol{\beta}$-coefficients, allowing the β_i:s to have different weights. If the weights are data-dependent and cleverly selected, as described by Zou [9], this gives the adaptive lasso the oracle property, meaning it is more consistent in its selection of variables. The $\boldsymbol{\beta}$-coefficients for adaptive lasso are derived from

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{alasso}} = \arg\min_{\boldsymbol{\beta}} \left\{ ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2 + \lambda \sum_{i=1}^{p} w_i |\beta_i| \right\} \tag{24}$$

where $\boldsymbol{w}$ is known as the *weight vector* and defined by

$$\boldsymbol{w} = (|\hat{\boldsymbol{\beta}}|)^{-\gamma}. \tag{25}$$

Here $\hat{\boldsymbol{\beta}}$ is an estimate of the $\boldsymbol{\beta}$-vector, derived from for instance the ordinary least squares solution in (11), and $\gamma > 0$ is a constant.

Adaptive Elastic Net is an extension of the elastic net, as well first presented by Zou, and Zhang, [1]. It was evolved from elastic net to compensate for its lack of the oracle property, and can be viewed as a combination of elastic net and adaptive lasso - adaptive lasso provides the oracle property while elastic net handles the case of correlated variables.

The adaptive elastic net uses the elastic net estimate, presented in (22), which can also be written as

$$\begin{aligned} \hat{\boldsymbol{\beta}}_{\boldsymbol{enet}} = (1+\lambda_2) \arg\min_{\boldsymbol{\beta}} \big\{ &||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2 + \\ &+ \lambda_2 ||\boldsymbol{\beta}||_2^2 + \lambda_1 ||\boldsymbol{\beta}||_1 \big\} \end{aligned} \tag{26}$$

combining (18) and (19). The Lagrange multipliers, λ_1 and λ_2, are related to α used in the elastic net by

$$\alpha = \frac{\lambda_2}{\lambda_1 + \lambda_2}. \tag{27}$$

The adaptive elastic net weights are obtained as in (25), with $\hat{\boldsymbol{\beta}}$ being the $\boldsymbol{\beta}$-estimate of ridge regression or elastic net. From this, the adaptive elastic net estimates, $\hat{\boldsymbol{\beta}}_{\boldsymbol{aenet}}$, are obtained from

$$\begin{aligned} \hat{\boldsymbol{\beta}}_{\boldsymbol{aenet}} = (1+\lambda_2) \arg\min_{\boldsymbol{\beta}} \Big\{ &||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2 + \\ &+ \lambda_2 ||\boldsymbol{\beta}||_2^2 + \lambda_1 \sum_{i=1}^{p} w_i |\beta_i| \Big\} \end{aligned} \tag{28}$$

where λ_1 is not necessarily the same one as in (22).

MARS stands for multivariate adaptive regression splines. Unlike the methods presented above, MARS uses bases $B(x)_i$, consisting of splines, to construct a model of high dimensional data. This makes (9) take the form

$$\hat{\boldsymbol{y}} = \sum_{i=1}^{n+k+1} \beta_i B(x)_i \tag{29}$$

where n, k and $B(x)_i$ will be presented below.

A spline is a piece-wise, continuous polynomial function that has continuous derivatives. A k:th order spline separated by n points, so-called *knot points*, at $t_1 < t_2 < ... < t_n$ can be described as a function $f : \mathbb{R} \longrightarrow \mathbb{R}$ that fulfills that f is a k:th degree polynomial on each of the intervals $(-\infty, t_1], [t_1, t_2], ..., [t_n, \infty)$. Furthermore, the spline has continuous derivatives of order $1, ..., k-1$ at its knot points. More about splines can be read about in Tibshirani [10].

Using the knot points, t_i, the basis functions, also called *truncated power functions*, are defined by

$$(x - t_i)_+^n = \begin{cases} 0 & x \leq t_i, \\ (x - t_i)^n & x > t_i \end{cases} \tag{30}$$

where the + sign indicates the positive part of $(x - t_i)^n$, i.e. $(x - t_i)_+^n = max\{(x - t_i)^n, 0\}$. Using these basis functions, the bases $B(x)_i$ in (29) can be built. The bases, also known as *truncated power bases*, can be either a constant, a truncated power function or a product of truncated power functions. The summation in (29) goes from 1 to $n + k + 1$ since each k:th

degree polynomial is defined by $k+1$ parameters, while the knots will divide the range of X into $n+1$ disjoint areas. However the continuity requirement of the polynomial at each knot point will place k constraints, reducing the $(n+1)(k+1)$ free parameters to be selected into $n+k+1$ free parameters.

Thus, both the locations of the knots and the β-values to be used in (29) are left to be determined. These parameters can be established by either minimization of the least-squares criterion in (11), or by using the truncated power basis in the least-squares criterion, see Friedman and Roosen [11]. The minimization is done with respect to β_i and the knot locations t_i.

The order of the spline, namely the degree of the polynomial it represents, is generally taken to be low, $k \leq 3$. More about this, as well as MARS in general, can be read about in Friedman and Roosen [11] and Friedman [12]. These papers also describe that MARS performs the best in models which are close to additive and that involve interactions in very few variables.

D. Cross Validation

One way to analyze how well a model will perform on a data set is to use cross validation, thoroughly described in Hastie et al. [2]. Some variable selection methods mentioned in Section II-C are using cross validation to extract optimal parameters. In cross validation, one artificial training set and one artificial test set are created by randomly dividing the given data set into some number of groups, let the number of groups be $K > 1$. In this report $K = 10$ will be used as default. The test set consists of one group, and the remaining $K-1$ groups will be defined as the training set. Cross validation estimates the model using the training data and validates the model's performance using the unseen test data. This procedure, with dividing the data into two different sets and then analyzing it, is repeated 10 times but with different groups as training and test set. Therefore it is also called 10×10-*fold cross validation*, which from now on will be referred to as *CV*.

E. Model Evaluation

The quantities used in this report for measuring the performances of the different methods consist of root mean squared error (RMSE), the coefficient of determination (R^2) and mean absolute error (MAE).

The **RMSE** is described by

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}{m}} \tag{31}$$

where y_i are the real measurements, $\hat{y}_i$ are the estimated values and m the number of measurements. The RMSE value describes the standard deviation of the residuals, meaning it explains how spread out the residuals are.

The $\boldsymbol{R^2}$ is defined as

$$R^2 = \frac{\sum_{i=1}^{m}(\hat{y}_i - \overline{y})^2}{\sum_{i=1}^{m}(y_i - \overline{y})^2} \tag{32}$$

where $\overline{y}$ is the mean of the vector of observed values. R^2 explains the fraction of variability in the data of the estimated values and the real values.

The **MAE** is given by the following formula

$$\text{MAE} = \frac{\sum_{i=1}^{m}|y_i - \hat{y}_i|}{m} \tag{33}$$

and describes, as the name indicates, the absolute value of the mean error.

To summarize, a good model would have a low RMSE, R^2 close to one and a low MAE. If the measurements are normalized, a high value implies a value close to one and a low value means a value close to zero.

III. Method

In this Section, the data set used in this project is presented. Also, the method for creating models of each variable selection method is described. Ridge regression, lasso, elastic net, adaptive lasso, adaptive elastic net and MARS were all implemented in R using different packages, see Rstudio [13]. The reason for using R for most of the methods was that built-in functions for adaptive lasso and adaptive elastic net does not exist in MATLAB. Furthermore, R is a suitable language for statistical mathematics. The OMP method was programmed in MATLAB, as described in Section II-C, since no working package for it was found in R.

A. Data Preprocessing

The data set used in this report was taken from *UCI Machine Learning Repository* [14], which is a source of data bases for machine learning. The used data set describes the number of violent crimes per 100 000 population in different folds in the U.S, which is also used as the observation vector of the data set, namely $\boldsymbol{y}$. This data was extracted in 2009. The number of violent crimes is determined by 127 independent variables that describe for example the number of police men per number of fold members, median family income, percent of housing occupied etc. The number of measurements is 1994. The data set, and all variable explanations, are thoroughly described in Appendix A.

All the observations and measurements in the data set are standardized and centered, as described in Section II-A. Out of the 127 independent variables in the data set, 122 were of numerical character and used in the project. In this report, the variables will be referred to as V1, V2, ..., V122, V123, with V123 being the observation. Furthermore, only 123 observations out of the 1994 in the data set were complete, meaning they had no missing values in their measurements. Therefore it was decided to use only these 123 observations in the project. In parallel to this, the same data set was used but with only the 50 first measurements. Thereby, all of the methods were implemented for both $m = 50$ and $m = 123$, respectively. The reason for this was to evaluate how the performance of each variable selection method varied when the number of predictors p were much larger than the number of measurements m, i.e. $p = 122$ and $m = 50$, compared to the case when the system was almost squared, i.e. $p = 122$ and $m = 123$. This means that two cases for the $\boldsymbol{X}$ matrix in (6) were considered: $\boldsymbol{X}$ as a 50×122 matrix and $\boldsymbol{X}$ as a 123×122 matrix.

B. R Packages

Ridge, Lasso, Elastic net and Adaptive Lasso were generated via the `train`-function, in the `caret`-package [15], as `glmnet`-objects [16]. For performing Adaptive Elastic net the `msaenet`-package [17] was used. The `earth`-package [18] was used to implement MARS.

C. Parameter Settings and β-coefficient Extraction

CV was used to obtain optimal parameter values for all methods besides OMP. Ridge regression, lasso and elastic net were implemented using (22). For **ridge regression**, $\alpha = 0$ was declared, and for **lasso**, $\alpha = 1$ was declared. For **elastic net** a sequence for α was set from 0 to 1 with 10 steps. For all three methods, λ was tested from 0.0001 to 1 with 1000 steps.

To perform **adaptive lasso** with the `train`-function, the weights had to be calculated first. That was done by running ridge regression again but now with the optimal lambda instead for a certain sequence. When the β-coefficients for ridge with optimal lambda was found, the weights for adaptive lasso could be determined by (25) for γ equal to 0.5, 1 and 2, as suggested in Zou [9]. After the weights were determined, the adaptive lasso could be performed. As for lasso, $\alpha = 1$ was set and λ was tested from 0.0001 to 1 with 1000 steps.

To perform **adaptive elastic net**, the penalty used in the initial estimation step was set to "ridge". As for elastic net, a sequence for α was set from 0 to 1 with 10 steps. λ selection criteria was set to "lambda.1se", which means that λ-value gets selected such that the error is within 1 standard error of the minimum.

For **MARS** the evaluation was done for the degree k, see (29), equal to 1, 2 and 3. The maximum number of bases was evaluated for 1,2,...,6.

For the methods that were returned as a `glmnet`-object, the β-coefficient for all the different λ values and the β-coefficient for the optimal λ, i.e. the final model, were extracted. The last data handling was done in MATLAB were the *variable importance* set was composed for lasso, elastic net and adaptive lasso. To establish the variable importance, the order in which the independent variables were selected by the different variable selection methods mentioned above was regarded. Some of the variables that were classified as important were selected several times, meaning they were also removed, and therefore they were defined as insecure selections and removed from the set of important variables.

For adaptive elastic net only the β-coefficient of the final model was extracted. For MARS the bases and respectively β-coefficient were extracted.

D. Model Evaluation Calculations

The model evaluation consisted of RMSE, R^2 and MAE, see Section II-E. These evaluation methods were built-in for the `train`-function and therefor already pre-calculated after each model extraction. The final model was determined by what λ that gave the best RMSE. For adaptive elastic net the RMSE and MAE were built-in, but the R^2 had to be calculated manually. To calculate R^2 for adaptive elastic net, a training set and a test set were declared as 90% respectively 10% of the measurements in the data set. Adaptive elastic net was then evaluated with the training set and predicted with the test set. Then (32) was used to calculate R^2. For OMP no model evaluation was done, since this there was no built-in function for this in MATLAB.

TABLE I
FOR $m = 50$: THE BEST AND FINAL TUNING PARAMETERS FOR THE METHODS RIDGE REGRESSION, LASSO, ELASTIC NET AND ADAPTIVE LASSO. * INDICATE THAT THESE METHODS DO NOT HAVE ANY WEIGHTS AND THEREFORE NO γ COULD BE CHOSEN.

Method	α	λ	γ
Ridge regression	0	1	*
Lasso	1	0.027	*
Elastic net	0.111	0.0991	*
Adaptive lasso	1	0.628	1

TABLE II
FOR $m = 123$: THE BEST AND FINAL TUNING PARAMETERS FOR THE METHODS RIDGE REGRESSION, LASSO, ELASTIC NET AND ADAPTIVE LASSO. * INDICATE THAT THESE METHODS DO NOT HAVE ANY WEIGHTS AND THEREFORE NO γ COULD BE CHOSEN.

Method	α	λ	γ
Ridge regression	0	1	*
Lasso	1	0.008	*
Elastic net	0.444	0.017	*
Adaptive lasso	1	0.027	0.5

TABLE III
For m = 50: FOR EACH OF THE VARIABLE SELECTION METHODS, EXCEPT OMP, THE TABLE SHOWS THE NUMBER OF SELECTED VARIABLES IN THE FINAL MODEL, THE RMSE, R^2 AND MAE.

Method	No. variables	RMSE	R^2	MAE
Ridge regression	121	0.142	0.793	0.115
Lasso	17	0.160	0.818	0.101
Elastic net	33	0.142	0.793	0.115
Adaptive lasso	8	0.160	0.698	0.132
Adaptive elastic Net	22	0.125	0.707	0.113
MARS	2	0.148	0.756	0.120

IV. RESULTS

All the values in the result Section are rounded to three decimals. The best tuning parameters, α, λ and γ, for the methods implemented with the `train`-function, except for MARS, are presented in Table I and Table II for $m = 50$ and $m = 123$, respectively.

Tables III and IV present the qualities and model evaluations of each of the variable selection methods, except for OMP, for $m = 50$ and $m = 123$ respectively. In the tables, the number of selected variables in the final model for each method is presented, as well as the evaluation quantities RMSE, R^2 and MAE. The RMSE values for all methods except OMP can also be visualised in Figure 5, where the difference in RMSE between $m = 50$ and $m = 123$ for each methods is emphasized.

TABLE IV
For m = 123: FOR EACH OF THE VARIABLE SELECTION METHODS, EXCEPT OMP, THE TABLE SHOWS THE NUMBER OF SELECTED VARIABLES IN THE FINAL MODEL, THE RMSE, R^2 AND MAE.

Method	No. variables	RMSE	R^2	MAE
Ridge regression	122	0.144	0.757	0.116
Lasso	16	0.123	0.801	0.090
Elastic net	20	0.123	0.803	0.090
Adaptive lasso	13	0.156	0.701	0.121
Adaptive elastic net	25	0.140	0.743	0.110
MARS	2	0.138	0.738	0.110

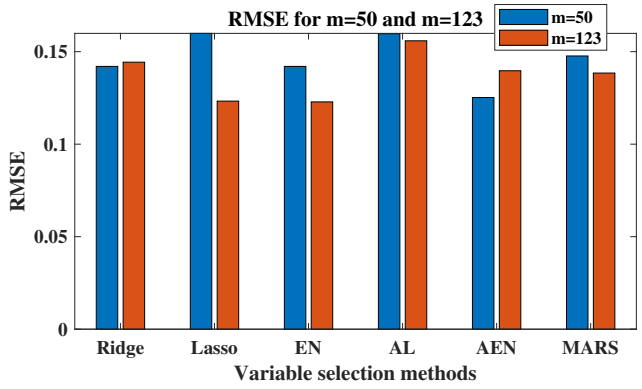


Fig. 5. The figure shows RMSE for, from the left, ridge regression, lasso, elastic net, adaptive lasso, adaptive elastic net and MARS. Blue bars (left) are for $m = 50$ and red bars (right) are for $m = 123$.

TABLE V
For m = 50: THE ORDER IN WHICH THE TOP TEN MOST IMPORTANT VARIABLES GOT SELECTED BY THE METHODS OMP, LASSO, ELASTIC NET AND ADAPTIVE LASSO.

Order	OMP	Lasso	Elastic net	Adaptive lasso
1	V51	V50	V9	V118
2	V91	V21	V49	V124
3	V119	V23	V50	V122
4	V82	V45	V52	V115
5	V1	V49	V51	V110
6	V103	V96	V56	V31
7	V3	V78	V8	V108
8	V5	V8	V21	V127
9	V90	V60	V23	V113
10	V33	V61	V45	V81

The variable importance, i.e. the ten variables that were extracted as the most important ones, for OMP, lasso, adaptive lasso and elastic net respectively are presented in Table V for $m = 50$ and Table VI for $m = 123$. Variables that at some time were removed from the set of selected variables, and then added again, have been removed from these tables due to the uncertainty of their importance. For the other methods it was not possible to extract the order of the selected variables.

Table VII and Table VIII display which variables were selected in the final model by different methods, by presenting corresponding β-coefficients, for both cases $m = 50$ and $m = 123$, respectively. OMP and ridge regression select all variables and therefore they are not relevant in this case. The variables that were selected by several methods are further visualized in box-plots in Figures 6 and 7, for $m = 50$ and $m = 123$.

TABLE VI
For m = 123: THE ORDER IN WHICH THE TOP TEN MOST IMPORTANT VARIABLES GOT SELECTED BY THE METHODS OMP, LASSO, ELASTIC NET AND ADAPTIVE LASSO.

Order	OMP	Lasso	Elastic net	Adaptive lasso
1	V51	V9	V52	V31
2	V91	V38	V9	V110
3	V33	V52	V23	V96
4	V47	V23	V38	V118
5	V6	V80	V21	V124
6	V56	V61	V80	V112
7	V107	V112	V68	V117
8	V112	V117	V112	V122
9	V28	V40	V61	V68
10	V90	V53	V48	V105

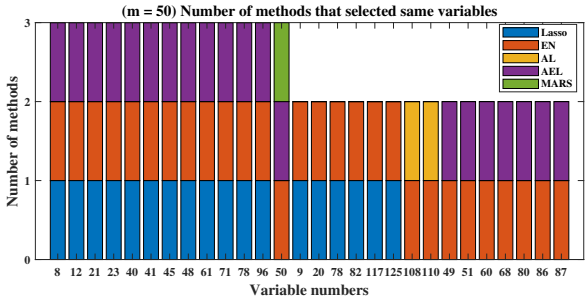


Fig. 6. *For m = 50:* Box-plot for all variables selected by more than one method. The x-axis shows the variable number i, for the variable Vi, and the y-axis the number of methods that selected the variable. All methods besides OMP and ridge regression are included here since they select all variables.

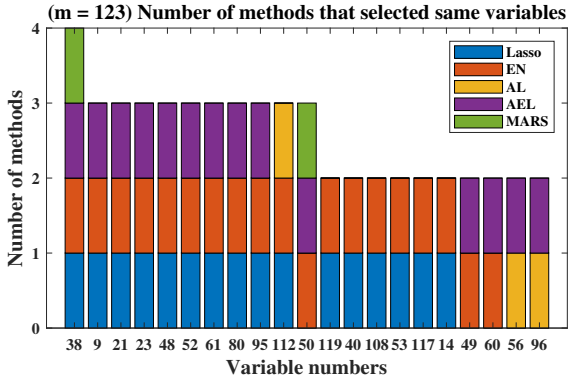


Fig. 7. *For m = 123:* Box-plot for all variables selected by more than one method. The x-axis shows the variable number i, for the variable Vi, and the y-axis the number of methods that selected the variable. All methods besides OMP and ridge regression are included here since they select all variables.

V. DISCUSSION

A discussion of the obtained results will be done in this Section. First, the resulting model evaluations will be discussed and compared. Then the selected variables of each method, and

TABLE VII
For m = 50: β-COEFFICIENTS FOR THE SELECTED VARIABLES OF THE VARIABLE SELECTION METHODS, FROM THE LEFT, LASSO, ELASTIC NET, ADAPTIVE LASSO, ADAPTIVE ELASTIC NET, MARS. * REPRESENTS COEFFICIENT OF A HINGE FUNCTION FOR VARIABLE V38.

Variable	Lasso	EN	AL	AEN	MARS
V8	0.285	0.089		0.104	
V12	0.005	0.01		0.031	
V21	-0.168	-0.154		-0.248	
V23	0.299	0.116		0.093	
V40	-0.018	-0.083		-0.079	
V41	0.121	0.084		0.036	
V45	0.065	0.074		0.117	
V48	0.010	0.120		0.209	
V50		-0.055		-0.045	5.051*
V61	0.238	0.094		0.108	
V71	0.239	0.147		0.160	
V78	-0.297	-0.227		-0.384	
V96	0.189	0.146		0.203	
V9	-0.003	-0.069			
V20	0.017	0.002			
V49		-0.060		-0.043	
V51		-0.039		-0.009	
V60		0.081		0.078	
V68		0.064		0.064	
V80		0.022		0.016	
V82	-0.108	-0.057			
V86		-0.018		-0.013	
V87		-0.017		-0.014	
V108		0.001	0.003		
V110		-0.001	-0.007		
V117	0.004	0.003			
V125	0.021	0.020			
V29		-0.002			
V30					5.051*
V31			-0.009		
V38		0.044			
V39				-0.113	
V44				0.035	
V52		-0.005			
V85		-0.012			
V94		0.030			
V95		0.040			
V113			-0.004		
V115			0.008		
V118			0.001		
V124			-0.001		
V127			0.003		

TABLE VIII
For m = 123: β-COEFFICIENTS FOR THE SELECTED VARIABLES OF THE VARIABLE SELECTION METHODS, FROM THE LEFT, LASSO, ELASTIC NET, ADAPTIVE LASSO, ADAPTIVE ELASTIC NET, MARS. * REPRESENTS COEFFICIENT OF A HINGE FUNCTION FOR VARIABLE V38. ** TWO DIFFERENT HINGE FUNCTIONS WITH THE SAME SELECTED VARIABLE.

Variable	Lasso	EN	AL	AEN	MARS
V38	0.169	0.172		0.066	0.470*
V9	-0.162	-0.112		-0.032	
V21	-0.079	-0.095		-0.142	
V23	0.297	0.223		0.054	
V48	0.183	0.196		0.114	
V50		-0.058		-0.081	-0.534, 1.093**
V52	-0.129	-0.119		-0.104	
V61	0.172	0.141		0.008	
V80	0.120	0.114		0.012	
V95	0.095	0.097		0.026	
V112	0.001	0.001	0.001		
V14	-0.043	-0.055			
V40	-0.164	-0.155			
V49		-0.042		-0.080	
V53	0.105	0.083			
V56			0.136	0.008	
V60		0.027		0.002	
V96			0.276	0.004	
V108	0.001	0.001			
V117	0.004	0.004			
V119	-0.116	-0.110			
V17				0.266	
V31			-0.007		
V34				0.039	
V39				-0.009	
V44				0.018	
V45				0.032	
V46				0.069	
V47				0.054	
V51				-0.047	
V68				0.115	
V69				0.044	
V71		0.032			
V78				-0.068	
V107			0.002		
V109			0.001		
V110			-0.003		
V111			0.001		
V113			-0.001		
V115			0.007		
V118			0.001		
V124			-0.002		
V127			-0.001		

the importance of these variables, will be analyzed. Further, robust variable selection and how the results from the different methods could be combined are considered, followed by some ideas on what future studies could be done in the topic of variable selection.

A. Model Evaluation

Since the optimal λ was decided by what model had the lowest RMSE, except for adaptive elastic net, this will be the most important aspect of the model evaluation. Though it is said that ridge regression shrinks coefficients towards

zero, it can still manage to shrink variables to exactly zero, as presented in Section II-C. Therefore, the number of selected variables can differ in the two cases of the size of m, which can be seen in Table III and Table IV. The RMSE improved for all methods, except for adaptive elastic net and ridge regression, as the number of measurements increased, see Figure 5. Lasso was the method that improved the most, according to RMSE, from $m = 50$ to $m = 123$.

When the number of measurements increases, the size of the training set also increases, see Section II-D. Therefore it is safe to say that the performance of the methods using CV should improve as m increases. The difference between the different measurement sizes for elastic net was not as big as for lasso, which according to the theory in Section II-C is true. Elastic net should outperform lasso when m decreases as stated in Zou and Hastie [8]. There is no good answer to why adaptive elastic net performed better for $m = 50$ than for $m = 123$, when comparing RMSE. Especially since adaptive elastic net can be seen as a combination of elastic net and adaptive lasso, and both these method performed better for $m = 123$.

In the case $m = 50$, elastic net and ridge regression got the same RMSE, and as seen in Table I elastic net selects α closer to 0 and therefore closer to ridge as described in Section II-C. On the other hand, when instead $m = 123$, elastic net and lasso got the same RMSE, and elastic net selects an α closer to lasso's $\alpha = 1$ than before, see Table II. It seems that elastic net tends to the better performing method between ridge and lasso in the different cases. For $m = 50$, adaptive elastic net performed best considering RMSE, meanwhile lasso and adaptive lasso performed the worst. Lasso, on the other hand, had the highest value of R^2 and the lowest value of MAE, while adaptive lasso got both the worst R^2 and MAE.

When the number of measurements increased to $m = 123$, lasso and elastic net achieved very similar results, as seen in Table IV, and both these methods clearly performed the best considering all three evaluation methods. Although, lasso selected fewer variables than elastic net, and hence got a more sparse model, which is desirable. Least performing method was adaptive lasso, that got the least satisfying result in all three evaluation methods.

B. Variable Selection and Importance

The variable importance, i.e. the order in which the variables were selected, by OMP, lasso, adaptive lasso and elastic net respectively, is presented in Table V for $m = 50$ and Table VI for $m = 123$. The table contains the ten variables that for each method were selected first, and thereby are the top ten most important ones.

Starting with $m = 50$, it can be concluded that V50, V21, V23, V49 were selected amongst the ten most important variables of both lasso and elastic net. V50, that was selected by the top three important variables by both lasso and elastic net, and the description of this variable can be seen in Table IX. This variable was also selected as important by MARS, see VII. Although, none of the variables selected as the top ten important ones by adaptive lasso were common with the ones selected by OMP, lasso and elastic net. OMP has selected only one variable as the top ten important ones in common with another method, namely $V51$ with elastic net.

TABLE IX
VARIABLE DESCRIPTIONS OF SOME OF THE VARIABLES THAT GOT SELECTED BY SEVERAL METHODS.

Variable name	Variable description
V50	Percentage of kids in family housing with two parents"
V112	Percent of police that are african american
V9	Percentage of population that is caucasian
V21	Percentage of households with investment / rent income in 1989
V38	Percentage of people 16 and over, in the labor force, and unemployed
V23	Percentage of households with public assistance income in 1989

The reason for the diversity in the selection of important variables across the methods could be due to the large number of predictors, combined with the few number of measurements. Consequently, it is of interest to compare the results above with the ones obtained for a larger number of measurements, namely $m = 123$.

In Table VI it is clear that for $m = 123$, V9, V38, V52, V23, V80 and V61 were selected by both lasso and elastic net as the top ten most important ones, while V117 was commonly selected by lasso and adaptive lasso and V68 by elastic net and adaptive lasso. The only variable selected by all four methods as one of the most important ones was V112. Also the top four selected variables of both lasso and elastic net, V9, V21, V38, V23, agrees. The mentioned variables, as well as V112, thus seem to be of importance, and the meaning of these variables can be seen in Table IX. All variable descriptions can be seen in the Appendix A.

As demonstrated by Tables V and VI, there are contrasts in the number of identically selected variables as the ten most important ones for the methods OMP, lasso, elastic net and adaptive lasso when comparing between $m = 50$ and $m = 123$. OMP selected V51, V91, V33 and V90 for both data set sizes. For lasso, V61 and V23 were selected for both $m = 50$ and $m = 123$. Elastic net selected V52, V9, V21, V23 for both number of measurements while adaptive lasso selected V118, V124, V122, V110, V31 both times. Thus, OMP has four, lasso two, elastic net four and adaptive lasso five variables that were selected for both numbers of measurements used in this project. Furthermore it is of interest to observe that the variable selected by all methods among the top ten most important ones, namely V112, for $m = 123$ was selected by none of the methods for $m = 50$. This is one indicator of that the consistency across variable selection methods increase when the number of measurements available increases.

Furthermore it can be established that adaptive lasso has selected nearly a completely different set of variables, compared to lasso, elastic net and adaptive elastic net, which can be visualised in Tables VII and VIII. This could be due to the implementation of adaptive lasso in R, where the weight vector in (25) was estimated outside of the built-in package. Moreover, the weight vector was determined using rigde regression, meaning the result of ridge regression has an impact on the weight vector and thereby the performance of adaptive lasso. The difference in selected variables of adaptive lasso could also be depending on this particular data set used.

MARS only selected two variables both for $m = 50$ and $m = 123$ to use in its final model, though it still performs averagely amongst the presented methods according to tables III and IV. As known from [11], MARS performs the best in models with very few variables being important. Therefore it is reasonable that MARS selects very few variables as important, which was also seen in the result of the study of Lima et al. [3].

In total, the variables selected by each method for $m = 50$ is presented in Table VII. Numerous of the variables selected by elastic net and adaptive elastic net are in common, as well as for lasso, while the ones selected for adaptive lasso were almost never the same as for the other methods. The same pattern can be seen in Table VIII for $m = 123$. Some variables where selected by more than two variable selection methods for both $m = 50$ and $m = 123$ - these where V21, V23, V48, V50 and V61. Thereby, these variables seem to be of major importance in the given data set.

Comparing the resulting number of selected important variables by the methods used in this report, to the number of variables selected by the methods used in the similar study by Lima et al. [3], it can be concluded that they are similar. In Lima et al., ridge regression as well selected almost all variables out of the 337 ones. Furthermore, elastic net and lasso selected similar amounts of important variables to each other for the data set used in that study, while MARS only selected two variables, exactly as seen in this project. Though, in the study by Lima et al., adaptive elastic net only selects three variables, thus creating a much more sparse model than seen in this report. Lima et al. did not implement adaptive lasso or OMP. The comparison with this study, however, gives an indication of how some of the methods perform similarly on the two different data sets, while others differ. This naturally has to do with what data set we are looking at, and what properties it has.

C. Robust Variable Selection and Combining Results Across Methods

As stated throughout the entire report, one fundamental aim of variable selection is to detect what independent variables that are the most important for the observation. Thus, it is of great interest to try to derive this information from the results presented above. First of all, the concept of robust variable selection will be analyzed, meaning what variables that will get selected even though small changes of the implementations are made. These changes could for example include changing the set measurements in the train and test set described in Section II-D.

In this project, it is therefor of interest to analyze what variables that were selected for both sizes of the data set, namely $m = 50$ and $m = 123$. The predictors selected for both these sizes could be viewed as the more robust ones, since they seem to have a major impact on the observation both when $p > m$ as well as $p < m$. As described in Section V-B, the methods OMP, lasso, elastic net and adaptive lasso had four, two, four and five, respectively, variables that were selected as the top ten important ones for both numbers of measurements. Thus it could be relevant to assume these are robust variables for each of the methods.

Another point of view for extracting what variables that are the most important can be done by examining which variables that got selected by several variable selection methods, as demonstrated in Figure 6 and Figure 7. These variables should seemingly have an indispensable impact on the observations. How many methods that have to select a certain variable, for it to be possible to draw the conclusion that the variable is important, is hard to define. However, for example in the case $m = 123$, the variable V38 got selected by all the methods except by adaptive lasso, as presented in Figure 7, cannot be neglected. A crucial impact on the result of which variables get selected is due to how the CV splits the observations into the training set and test set. Depending on how the measurements get grouped together, the model extraction for the different methods, and thereby the resulting selected variables, as well as the model evaluations, can vary.

Further, the disparity in performances of the different variable selection methods has to be taken into consideration. It could be applicable to put more confidence in the methods that for a certain data set result in a lower RMSE and MAE, and a higher R^2. Although, these evaluation methods obviously do not represent the entire performance of a variable selection method on a specific data set. Nevertheless, the methods all perform differently on the same data set used in this report.

It is essential to observe that the different variable selection methods are performing well on different kinds of data sets. As described in the Background Section II, lasso and adaptive lasso perform well by selecting only a small number of important variables, while ridge regression, elastic net and adaptive elastic net are able to select groups of correlated variables. MARS performs well when the set of real important variables is small. OMP is good at giving an order of the variable importance. Thus, since it is usually unknown what of these properties that are applicable for the data set to be analyzed, it is a difficulty to conclude what variable selection method that is the most appropriate to use for the current data set.

For the data set used in this project it is clear from Tables III and IV that elastic net and adaptive elastic net in both cases selects more variables than lasso as well as adaptive lasso. This could be due to some grouping of variables, i.e there exists correlated sets of variables in the data set. In these cases, elastic net and adaptive elastic net tend to select all of these correlated variables. Inspecting the results of the evaluation methods for MARS in Tables III and IV it is demonstrated

that MARS does not perform much more unsatisfactory than the other methods, although it only selects two variables for both $m = 50$ and $m = 123$. For $m = 50$ it has a lower RMSE than both lasso and adaptive lasso. In the case with $m = 123$, MARS has lower RMSE than adaptive elastic net, adaptive lasso and ridge regression and also lower MAE than adaptive lasso and ridge regression. One reason for this could be that MARS selects variables that are particularly important. This would lead to the reasoning that there are very few variables in the data set that are of major importance, which would explain why MARS does not have extremely bad values for RMSE, R^2 and MAE compared to the other methods, although only using two variables in the final model.

D. Ideas for Future Studies

For future studies within variable selection in high-dimensional data it can be of interest to analyze methods for telling if a variable is important and robust. Ideas that came up during this project around this topic, but could not be realized because of the time plan, are the following:

- Say that a variable selection method has been implemented on a high-dimensional data set and resulting in a set of important variables. If then the variables are removed one at a time, one can study how the new model affects the RMSE. It should then be possible to conclude if the variable was as important as thought. If the RMSE gets much higher after a variable is removed, it should be considered more important than the others.
- It was also interesting to see which variables that were robust, in other words got selected for both $m = 50$ and $m = 123$. If one takes this a step further, this could be analyzed for numerous sizes of the same data set. Consider that the analyzed data set decreases with one measurement at a time, and at each new size of the data set an arbitrary variable selection method is used for estimating the model. If then a variable gets selected for several sizes of the data set, it should be viewed as important and robust.

VI. Conclusion

All of the implemented variable selection methods performed differently from each other on the high-dimensional data set used in this report. They differed both in what variables were used in the final model and in the results of the evaluation methods. Furthermore, the methods for which the variable importance was extracted selected a different order of important variables. The diversity in the final models of the different methods imply the difficulty in determining what variables are actually important for a data set. Nevertheless, we can assume which variables that are important by concluding that variables selected by several methods are more likely than others to be of importance in the final model, like for example variable $V38$. The combination of results across methods can therefore provide further insight in what variables that are of importance for a specific data set.

Appendix A
Description of Data set

Appendix B
Processed Data set

Acknowledgment

The authors would like to thank Prakash Borpatra Gohain for his indomitable support and guidance throughout the entire bachelor thesis project and Magnus Jansson for his supervision and feedback.

References

[1] H. Zou and H. Zhang, "On the adaptive elastic-net with a diverging number of parameters," *Project euclid*, vol. 37, no. 4, pp. 1733–1751, Aug. 2009.
[2] T. Hastie, R. Tibshirani, and M. Wainwright, *"Statistical Learning With Sparsity"*. Chapman and Hall/CRC, Hoboken, London., May 2015, vol. 1st Edition, no. 9780367738334.
[3] E. Lima, P. Davies, J. K, F. Lovatt, and M. Green, "Variable selection for inferential models with relatively highdimensional data: Between method heterogeneity and covariate stability as adjuncts to robust selection," *Scientific Reports*, no. 8002, May 2020.
[4] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis*, Dec. 2012, vol. 5th edition.
[5] R. Sridharan, "Chapter 3 - linear regression," MIT, Massachusetts, 2015, lecture handouts in course S085 Statistics for Research Projects.
[6] T. T. Cai and L. Wang, "Orthogonal matching pursuit for sparse signal recovery with noise," *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 4680–4688, Jun. 2011.
[7] C. E. Ginestet, "Regularization: Ridge regression and lasso," Boston University, Massachusetts, 2013, lecture handouts in course CAS MA 575: Linear Models.
[8] H. Z. T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society Series B (Statistical Methodology)*, vol. 67, pp. 301–320, Mar. 2005.
[9] H. Zou, "The adaptive lasso and its oracle properties," *Journal of the American Statistical Association*, vol. 101, pp. 1418–1429, Jan. 2012.
[10] R. Tibshirani, "Smoothing splines," Carnegie Mellon University, Pennsylvania, 2014, lecture handouts in course Advanced Methods for Data Analysis (36-402/36-608).
[11] J. H. Friedman and C. B. Roosen, "An introduction to multivariate adaptive regression splines," *Sage Journals*, vol. 4, pp. 197–217, Sep. 1995.
[12] J. H. Friedman, "Multivariate adaptive regression splines," *The Annals of Statistics*, vol. 19, no. 1, pp. 1–67, Mar. 1991.
[13] (2021, Apr.) Rstudio. Program used for implementation of methods. [Online]. Available: https://www.rstudio.com/
[14] M. Redmond. (2009, Jul.) Communities and crime data set. Philadelphia, USA. Data set used in report. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Communities+and+Crime
[15] Max Kuhn, Jed Wing, Steve Weston, Andre Williams, Chris Keefer, Allan Engelhardt, Tony Cooper, Zachary Mayer, Brenton Kenkel, R Core Team, Michael Benesty, Reynald Lescarbeau, Andrew Ziem, Luca Scrucca, Yuan Tang, Can Candan, Tyler Hunt. (2020, Mar.) Package 'caret'. [Online]. Available: https://cran.r-project.org/web/packages/caret/caret.pdf
[16] Jerome Friedman, Trevor Hastie, Rob Tibshirani, Balasubramanian Narasimhan, Kenneth Tay, Noah Simon, Junyang Qian. (2021, Feb.) Package 'glmnet'. [Online]. Available: https://cran.r-project.org/web/packages/glmnet/glmnet.pdf
[17] Nan Xiao, Qing-Song Xu. (2019, May) Package 'msaenet'. [Online]. Available: https://cran.r-project.org/web/packages/msaenet/msaenet.pdf
[18] Stephen Milborrow, Trevor Hastie, Rob Tibshirani, Alan Miller, Thomas Lumley. (2020, Oct.) Package 'earth'. [Online]. Available: https://cran.r-project.org/web/packages/earth/earth.pdf

Compression and Distribution of a Neural Network With IoT Applications

Hannes Backe and David Rydberg

***Abstract*—In order to enable deployment of large neural network models on devices with limited memory capacity, refined methods for compressing these are essential. This project aims at investigating some possible solutions, namely pruning and partitioned logit based knowledge distillation, using teacher-student learning methods. A cumbersome benchmark teacher neural network was developed and used as a reference. A special case of logit based teacher-student learning was then applied, resulting not only in a compressed model, but also in a convenient way of distributing it. The individual student models were able to mimic the parts of the teacher model with small losses, while the network of student models achieved similar accuracy as the teacher model. Overall, the size of the network of student models was around 11% of the teacher. Another popular method of compressing neural networks was also tested - pruning. Pruning the teacher network resulted in a much smaller model, around 18% of the teacher model, with similar accuracy.**

***Sammanfattning*—För att möjliggöra användning av stora neurala nätverksmodeller på enheter med begränsad minneskapacitet krävs raffinerade metoder för komprimering av dessa. Detta projekt syftar till att undersöka några möjliga lösningar, nämligen pruning och partitionerad logit-baserad knowledge distillation, med hjälp av teacher-student-träning. Ett stort riktmärkesnätverk utvecklades och användes som referens. En speciell typ av logit-baserad teacher-student-träning tillämpades sedan, vilket inte bara resulterade i en komprimerad modell utan också i ett smidigt sätt att distribuera den på. De enskilda student-modellerna kunde efterlikna delar av teacher-modellen med små förluster, medan nätverket av student-modeller uppnådde ungefär samma noggrannhet som teacher-modellen. Sammantaget uppmättes storleken av nätverket av student-modeller till cirka 11 % av teacher-modellen. En annan populär metod för komprimering av neurala nätverk testades också - pruning. Pruning av teacher-modellen resulterade i en mycket mindre modell, cirka 18 % av teacher-modellen i termer av storlek, med liknande noggrannhet.**

***Index Terms*—Machine Learning, Neural Network, IoT, Compression, Pruning, Knowledge Distillation (KD), Distributed Machine Learning (DML)**

Supervisors: Ragnar Thobaben

TRITA number: TRITA-EECS-EX-2021:186

I. Introduction

The term *neural network* has become a real buzzword in our everyday life, and rightfully so. The application of these algorithms, for both industrial and domestic usage, greatly improves efficiency in decision making processes. Paired with the large amount of data humanity has been collecting during the recent years, it constitutes an important tool for learning and inference based on these large data sets.

The rapid development of hardware power has inflated the complexity of neural networks [1], which creates fundamental bottlenecks in some areas of application. One of these areas is the implementation of neural network algorithms on small devices, such as smartphones and IoT (Internet of Things) equipment. Devices which have limited memory capacity and computation power.

IoT networks commonly utilize a central processing unit, alternatively a cloud based solution, to perform calculations and to store data [2]. This raises several concerns in terms of potential privacy breaches as well as failing to take advantage of the gain of parallel computation. An IoT network, which could perform these calculations distributed on the devices, commonly known as distributed edge computation, could reduce the risk of data breaches while also performing the tasks in parallel, thus lessening the computation time. In addition, a cloud based solution brings latency concerns due to high communication costs. Distributed edge computation provides an alternative approach which could minimize these.

In order to deploy complex models on devices with high memory constraints, computer scientists have developed methods to compress such networks. One of the most common techniques is pruning, which reduces both the memory required as well as the runtime for the algorithm. This is an effective method for reducing network energy consumption [3]. Another developmental method is knowledge distillation (KD) [4], which describes a procedure of transferring knowledge between networks. By using knowledge distillation to transfer knowledge from a large model to a smaller, this method could in practise be considered a compression technique.

In this project these techniques were implemented in order to address the enabling of distributed edge computing in IoT networks. To accomplish this, a neural network was developed to classify presence in an apartment using IoT sensor data provided by KTH Live-In Lab. The model was then compressed and distributed by employing a variant of teacher-student learning [4]. In addition, the model was pruned to weigh the performance of the knowledge distillation method against a more reputable compression technique.

II. Deep Neural Networks

In this section, a brief introduction to relevant information regarding neural networks is given. The underlying structure and mathematics is of highest importance in regards to the theories on which the used methods are based on.

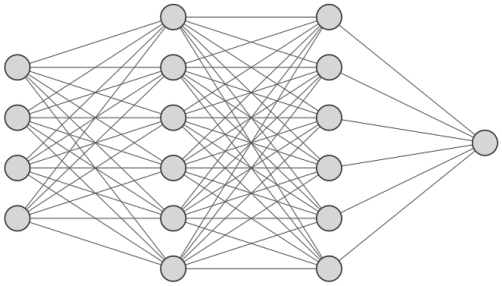


Fig. 1. Illustration of a simple, fully connected, neural network with two hidden layers and a single output node.

A. Architecture and Elementary Overview

A deep neural network is a mathematical model framework, which is inspired by the human brain composition. Neurons of the brain are represented by nodes in the model, which are assembled into layers, as illustrated in Fig. 1. The intent of the nodes is to transfer signals between each other, thus passing information through the layers. The ultimate goal of such a model is to reproduce some unknown function

$$\mathbf{y} = \mathbf{f}^*(\mathbf{x}) \quad . \tag{1}$$

A layer can be modelled in terms of its nodes, which in turn take an input vector and apply an *activation function* before outputting a value, according to the schematic equation

$$\mathbf{n}(\mathbf{g}(\mathbf{x})) : \mathcal{R}^N \rightarrow \mathcal{R}^1 \quad , \tag{2}$$

where $\mathbf{n}$ is the functionality of the node and $\mathbf{g}$ represents the applied activation function.

Alternatively, the process of feeding information through the model can be described on a layer level by the equation [5]

$$\mathbf{h}^{(i)} = \mathbf{g}^{(i)} \left(\mathbf{W}^{(i)T} \mathbf{h}^{(i-1)} + \mathbf{b}^{(i)} \right) \quad , \tag{3}$$

where $\mathbf{g}^{(i)}$ corresponds to the activation function applied to layer i, $\mathbf{W}^{(i)T}$ is the *weight matrix* operating on the previous layer output, $\mathbf{b}^{(i)}$ represents the *bias vector* and $\mathbf{h}^{(i)}$ is the computed output. The activation function $\mathbf{g}^{(i)}$ is considered a *hyperparameter*, a term that is further elaborated in Section II-B, and is adjusted or chosen accordingly. Common activation functions include *Rectified Linear Unit*, ReLU [6]

$$f_{ReLU}(x) = \max(0, x) \quad , \tag{4}$$

which is applied element wise to each vector component. Another activation function, commonly used in the final output layer, is the *Sigmoid* function [6]

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad , \tag{5}$$

which is also applied element wise. These are both activation functions that were used during the course of this project.

The process of passing data through the network, called *forward propagation*, is complemented by *backward propagation* methods, which control the parameter adjustments during the learning process. Training the model aims to assign values to the weight and bias tensors in order to approximate the function $\mathbf{f}^*(\mathbf{x})$ for all given inputs of $\mathbf{x}$. An indicator of how well the model performs in each learning step, *batch*, is thus required. This measurement is evaluated through a *loss function*, which determines the amount of penalization on the weights and biases. Loss functions are highly dependent on the intents and tasks of the model and classifying models are often equipped with a variant of *crossentropy* loss measurement, similar to

$$L = -\sum_{i=1}^{n} y_i \cdot log\left(\hat{y}_i\right) \quad . \tag{6}$$

For non-classifying neural networks, other loss functions are better suited, such as the mean squared error function

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2 \quad , \tag{7}$$

where y_i and $\hat{y}_i$ represents the true value and predicted value respectively. The objective of back propagation is to optimize the parameters of the neural network in order to minimize the loss function. Similar to other numerical algorithms, *gradient decent* is used to recursively update the parameters in order to find a good minimum of the loss function.

1) Binary Classification: A binary classifier is a classifying model with two possible outputs. This can be modelled either with one or two output nodes. In the case of one output node, the *soft label* is rounded off to the closest *hard label*, i.e either zero or one. If the training data is well balanced, the *accuracy* measurement is appropriate and simply utilized as a performance metric. It is defined according to

$$A = \frac{\#correct\,classifications}{\#total\,classifications} \quad . \tag{8}$$

In case of binary classification, Equation (6) is not well suited. Instead, a *binary crossentropy* loss function, an average version of crossentropy, is commonly applied according to

$$L = -\frac{1}{N} \sum_{i=1}^{N} y_i \cdot log\left(\hat{y}_i\right) + (1 - y_i) \cdot log\left(1 - \hat{y}_i\right) \quad , \tag{9}$$

where N is the output size.

Furthermore it is essential to apply an appropriate activation function to the final output node that enables normalization of the outputs and mapping them to a probability between zero and one. As noted in Equation (5), this is precisely what the Sigmoid function accomplishes. In this report binary cross entropy loss (9) is used during the optimization of the baseline teacher model and mean squared error loss (7) during the optimization of the student models.

2) Reducing the Risk of Overfitting: A complex architecture of a neural network often results in *overfitting* on the training data. This occurs when the weights and biases are tuned to accurately perform well on the training data, while the model is not able to generalize its task to previously unseen data. In order to avoid this phenomena, noise can be introduced

into the training process. One way of doing so is to deploy *dropout layers* and *regularizing functions*. Dropout layers randomly ignore a specific percentage of nodes during each training batch, effectively introducing random variations into the process [7]. Regularizing functions on the other hand, restrict the parameters by penalizing larger parameters during back propagation [8]. By adding an additional penalty term to the loss function the new loss, L', is denoted [9]

$$L' = L + \lambda\Omega(\theta) \quad , \tag{10}$$

where L is the original loss, λ is the regularization factor and

$$\Omega(\theta) = \|\theta\|^2 = \sum_{i=1}^{n} \theta_i^2 \quad . \tag{11}$$

B. Hyperparameter Optimization

Parameters of a neural network model that are selected by the creator are considered to be *hyperparameters*. Tuning these manually is immensely time consuming due to the large amount of hyperparameter permutations in combination with a slow training process. However, several algorithms have been proposed and implemented to reduce the required work load, such as *Random Search* [10] and *Bayesian optimization* based algorithms [11], which was used in this report.

Bayesian optimization is based on a Gaussian process and aims at, as all optimizers, finding the minimum of some bounded unknown function. Unlike a Random Search which generates random samples of hyperparameters that are evaluated and ranked, a Bayesian optimization algorithm establishes a probabilistic model in the hyperparameter space, which newly generated hyperparameter permutations are based upon. This is performed by constructing a series of observations, denoted [12]

$$\begin{gathered} S = \{\mathbf{x}_n, y_n\}_{n=1}^{N} \\ \text{where } y_n \sim \mathcal{N}(f(\mathbf{x}_n), \nu) \quad . \end{gathered} \tag{12}$$

Furthermore the method relies on the choice of an *acquisition function*. The purpose of the acquisition function is to determine which hyperparameters should be picked for evaluation in the next time step. The acquisition function used in this project is [12]

$$\mathbf{x_t} = \underset{\mathbf{x}\in D}{\operatorname{argmax}}\, \mu_{t-1}(\mathbf{x}) - \beta^{1/2}\sigma_{t-1}(\mathbf{x}) \quad . \tag{13}$$

This equation constitutes the GP-UCB acquisition function, which is a denotation for *Gaussian Process - Upper Confidence Bound* [12]. Equation (13) depends solely on the observed predictive mean function $\mu(\mathbf{x})$ and variance function, $\sigma(\mathbf{x})$, along with a tuning parameter, β, controlling the relation between exploration and exploitation of the algorithm.

III. Compression and Distribution

The theory behind the compression and distribution methods used in this project are treated in this section. As discussed in Section I, these constitute the foundation of this project. While the precise algorithm of pruning is discussed, the distribution technique is merely provided as a general framework and details are to be found in Section IV-C.

A. Pruning

One of the most famous compression techniques is *pruning*. Pruning algorithms reduce a neural network's size by 'cutting connections' between nodes or 'cutting' the nodes themselves. This report will merely consider pruning of connections between the nodes. In the TensorFlow framework [8] this is done by zeroing unnecessary weights.

The process of pruning in TensorFlow is described as follows [13]: The used weights in the forward execution of each layer is identified. The weights are then sorted by size and the smallest weights are set to zero until the specified *sparsity* is reached. During back propagation the previously mentioned identified weights are not updated. The process above is executed gradually with an increasing sparsity s_t as in the equation [13]

$$\begin{gathered} s_t = s_f + (s_i - s_f)\left(\frac{t - t_0}{n\Delta t}\right)^3 \quad , \\ t \in \{t_0, t_0 + \Delta t, ..., t_0 + n\Delta t\} \end{gathered} \tag{14}$$

where s_i is the initial sparsity, s_f the final sparsity, n the number of pruning steps, t_0 the initial pruning step and Δt the pruning frequency.

If applied correctly pruning can reduce the size of a neural network significantly. For example, researchers were able to reduce the number of non-zero parameters in AlexNet by a factor of 9 without any loss of accuracy [14].

B. Knowledge Distillation

A deep neural network can still contain millions of parameters after pruning. Consequentially such networks are hard to fit on memory-constrained devices. One way to bypass this problem is to distill knowledge from the large model, called the teacher, onto a smaller model, called a student. This approach of knowledge distillation is called teacher-student learning [15]. In teacher-student learning, a large teacher network is trained to accurately perform its desired task. Then student models are trained on some information produced by the teacher to be able to mimic it while often being smaller in size.

Binary classification hard labels may however not be the optimal choice of training labels, even though it may seem intuitive. As discussed in [16], training on hard labels could result in the student model failing to generalize well on the data, since hard labels do not contain information of uncertainty in a classification. For instance, if the soft label is equal to 0.51, the binary classifier would classify the hard label as 1, while if the soft label is equal to 0.49, the hard label would be 0. One possible approach to address this problem is to simply train on the soft labels, although a different solution was tested in this project as explained in Section IV-C.

IV. Methodology

The methods used to accomplish the desired task - to build, compress and distribute a large neural network in order to enable parallel distributed edge computing, is presented in chronological order. A schematic image of the general workflow is provided in Fig. 2, while the details regarding the developmental processes are divided into several sections.

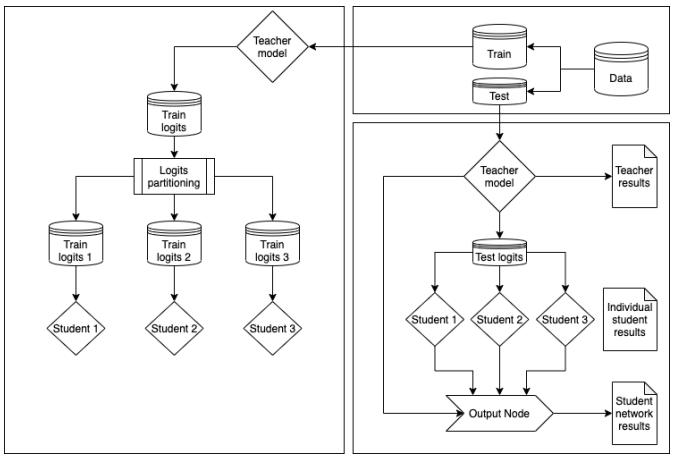


Fig. 2. Schematic workflow overview. The training processes of the models are illustrated in the left rectangle while the distribution of the teacher model is shown in the bottom right rectangle.

A. Data Pre-processing

The data used for classifying presence in the apartment has been provided by KTH Live-In Lab. The IoT sensors were located in three rooms in one apartment and motion detectors, indicating occupancy, were used as a reference for the true labels. The parameters used were luminosity, CO_2 concentration, temperature and air humidity. The placement of the sensors can be found in Table I. Measurements were provided on an hourly basis and each motion sensor had an output value between 0 and 1. In case all motion sensor outputs were equal to 0, the apartment was considered to be empty. If at least one motion detector returned 1, the apartment was considered to be occupied. In case some sensors returned a reading between 0 and 1 but none returned a reading equal to 1 the case was considered ambiguous and the measurement was removed from the data set.

In total 2636 data points were used, which were partitioned into 75% training and 25% testing data and normalized according to the standard score formula

$$\hat{X} = \frac{X - \mu}{\sigma} \quad . \tag{15}$$

B. Teacher Network Model

Due to the limited amount of training data, a modified Bayesian optimization algorithm was implemented, using *stratified cross validation*, with the intent to minimize fluctuations in the scores of each hyperparameter set. A subset of hyperparameters, from which the algorithm was allowed to generate new hyperparameter sets was defined according to Table II. Bold text indicates the final selected hyperparameters that were used in the model.

TABLE I
Data Parameters

Parameter	Sensor location
Luminosity	kitchen, bathroom, living room
CO_2 concentration	kitchen, bathroom, living room
Temperature	kitchen, bathroom, living room
Air humidity	bathroom, living room

TABLE II
Bayseian Optimization Parameter Options

Hyperparameter	Values
Nodes - layer 1	{256, 312, **412**, 512, 624}
Nodes - layer 2	{**64**, 112, 206, 256, 312}
Nodes - layer 3	{48, **64**, 128, 256}
Nodes - layer 4	{**10**, 12, 24, 56, 64}
Nodes - layer 5	{12, 15, **18**}
Dropout - layer 1	{0.001, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, **0.45**}
Dropout - layer 2	{0.001, 0.1, 0.15, **0.2**, 0.25, 0.3, 0.35, 0.4, 0.45}
Dropout - layer 3	{**0.001**, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45}
Bias Regularizer - layer 1	{0.001, 0.01, 0.05, **0.1**, 0.2, 0.3}
Bias Regularizer - layer 2	{0.001, 0.01, 0.05, **0.1**, 0.2, 0.3}
Bias Regularizer - layer 3	{0.001, 0.01, 0.05, 0.1, 0.2, **0.3**}
Bias Regularizer - layer 4	{0.001, **0.01**, 0.05, 0.1, 0.2, 0.3}
Learning rate (SGD)	{0.0005, **0.0008**, 0.001, 0.0015, 0.01}
Momentum	{0.5, 0.6, 0.7, 0.8, **0.9**}
Batch size	{10, 15, 20, 25, **30**, 35, 40}

A rather large model was required in order to perform pruning and partitioning. Thus a fairly high number of nodes were fed into the Bayesian optimization algorithm.

Five teacher models were trained for 400 *epochs*, using *early stopping* monitoring validation loss, with *patience* set to 30 epochs. Binary cross entropy (9) was chosen as the loss function. ReLU (4) was used as activation function for all layers except the output layer, where Sigmoid (5) was applied. The accuracies and losses during the training process are illustrated in Fig. 3. In addition all teacher models were pruned to a final sparsity of 95% using Tensorflow built in functionality, which has its roots in Equation (14). The distribution of the weights in four layers of the pruned and non-pruned teacher models can be found in Fig. 4.

C. Student Network Models

TABLE III
Student Models Architecture

Hyperparameter	Value
Nodes - layer 1	20
Nodes - layer 2	18
Bias Regularizer - layer 2	0.1
Dropout - layer 2	0.1
Nodes - output layer	6
Learning rate (Adam)	0.001

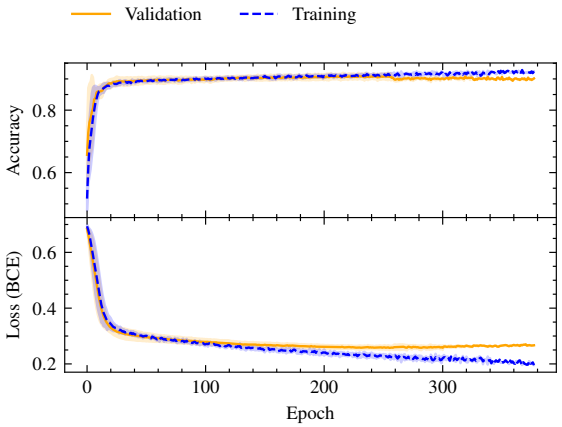


Fig. 3. Accuracy and loss during training of teacher model.

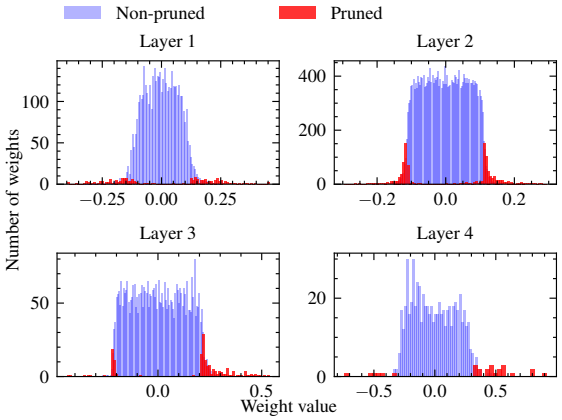


Fig. 4. Weight distributions of of the non-pruned and pruned teacher model.

In order to compress the teacher model, various parts of the teacher's knowledge were transferred to different student models. This approach has aspects similar to the work done in [16], but rather than using *ensemble learning* methods, a solution utilizing partitioned hidden layer logits training was used. In this case, the term *logits* refers to the raw output from each layer in the model. The intention of this approach to teacher-student learning is similar to the idea of training on soft labels, to not lose valuable information and to enable distribution of the teacher's knowledge between different students. The idea was that training different students on different labels would enable the individual student models to accurately reproduce the information they were trained on while still being considerably smaller than the teacher.

Three student models for each teacher were developed, with the goal of predicting the logits produced in the final hidden layer of the teacher model for every given input. This was performed by generating the logits produced by the teacher model. The generated logits were split into three disjunct sets, with six logits in each set, and this was then used as training data for the student models. The assignment of the logits for each student model was arbitrary.

The student model architecture is described in Table III and *Adam* was used as the optimizer for the student models, as opposed to SGD for the teacher model. Identical architectures were used for all student models, and hyperparameter tuning was based on manual testing, since the models were less complex compared to the teacher model.

The students were trained for 200 epochs. The training and validation losses of the individual student models during training are illustrated in Fig. 5 and 6. Finally the students were allowed to predict occupancy by feeding their outputs into the final output node of the teacher model.

Sizes of all the models were measured by saving the models as h5-files and zipping them.

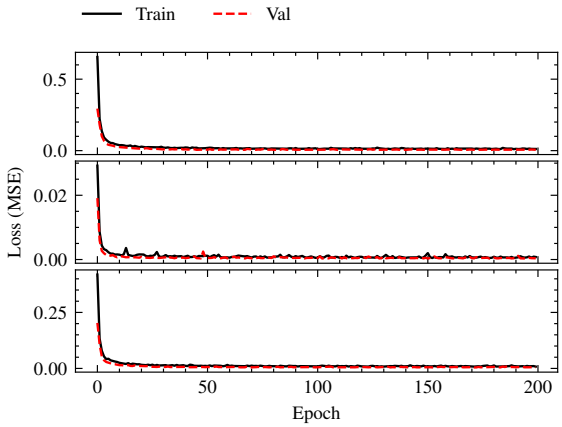


Fig. 5. Training and validation loss during training of student models.

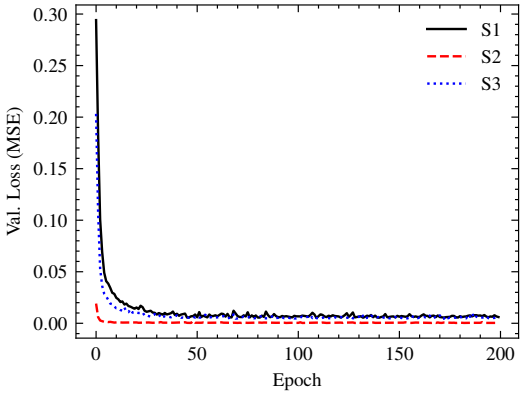


Fig. 6. Validation loss during training of student models.

V. Results

TABLE IV
Size, Accuracy and Loss of Teacher Model (Pruned Sparsity 95%)

Accuracy (Pruned) [%]	Loss (Pruned) [BCE]	Size (Pruned) [kB]
92.41 (92.41)	0.218 (0.236)	137.070 (24.746)
92.87 (92.11)	0.229 (0.231)	137.080 (24.809)
93.32 (92.26)	0.222 (0.230)	137.074 (24.767)
93.32 (92.56)	0.209 (0.221)	137.305 (24.798)
92.72 (91.96)	0.236 (0.236)	137.063 (24.820)

TABLE V
SIZE, ACCURACY AND LOSS OF NETWORK OF STUDENT MODELS

Accuracy [%]	Loss [BCE]	Size [kB]
91.65	0.219	15.361
92.26	0.224	15.370
92.26	0.225	15.358
92.41	0.205	15.335
92.11	0.242	15.381

TABLE VI
SIZE AND LOSS OF STUDENT MODELS

Student Model	Loss [MSE]	Size [kB]
$S1_1$	0.00498	4.676
$S1_2$	0.00102	4.684
$S1_3$	0.00741	4.674
$S2_1$	0.00364	4.683
$S2_2$	0.00313	4.682
$S2_3$	0.00298	4.678
$S3_1$	0.00629	4.675
$S3_2$	0.000798	4.679
$S3_3$	0.00969	4.677
$S4_1$	0.00690	4.669
$S4_2$	0.0195	4.666
$S4_3$	0.00351	4.673
$S5_1$	0.00296	4.681
$S5_2$	0.00688	4.686
$S5_3$	0.00359	4.687

In summary, the teacher model reached a test accuracy of 92.93 ± 0.394 %, a loss of 0.22 ± 0.01 and a size of 137.12 ± 0.104 kB. Pruning the teacher model to a final sparsity of 95% resulted in a test accuracy of 92.26 ± 0.237 %, a loss of 0.23 ± 0.006 and a size of 24.79 ± 0.031 kB. The network of student models reached a test accuracy of 92.14 ± 0.293 %, a loss of 0.22 ± 0.013 and a size of 15.36 ± 0.017 kB. Detailed results are presented in Table IV for the non-pruned/pruned teacher model and in Table V for the network of student models. Each row in the tables represents a cross-validation run. The performance results and sizes of individual student models in regards to mimicking the logits of the teacher model are displayed in Table VI. Sx_y refers to the student trained on logits set y of teacher x.

VI. DISCUSSION

Divided into several sub-processes, the results and methods are discussed in this section. Proposals for future studies are also made.

A. Teacher Model

The Bayesian optimization algorithm returned a model with a sufficient test accuracy as demonstrated by Table IV. The results indicate that the teacher generalizes well on test data although Fig. 3 suggests the network may be somewhat overfitted due to the growing discrepancy between the training and validation losses from around epoch 200 and forward.

The model was developed solely with the intention of maximizing the accuracy. A smaller network may have been sufficient depending on the model requirements and the data. Applying the same method for knowledge distillation as discussed in Section IV-C on a neural network developed with its size in mind may not yield the same compression results as in this report.

Combining our implementation of cross-validating the hyperparameters with Bayesian optimization for maximum accuracy with larger restrictions on the maximum memory requirements could result in a smaller teacher model with comparable accuracy. The size of a model is not the only parameter determining its performance. In other words, a larger model may not always result in higher accuracy and lower loss, due to phenomena such as overfitting.

The Bayesian optimization algorithm may also not be sufficient to determine the optimal parameters of the network.

Pruning the teacher network yielded positive results. The pruned networks occupied approximately 18 % of the memory of the original teacher models with no significant loss of accuracy.

B. Individual Student Models

The intent behind partitioning the logits before training, rather than training all students on all logits, was to minimize the size of the students. Ideally, this strategy enables the students to mimic different elements of the teacher, as opposed to some varieties of ensemble learning, where the aim of every student is to reproduce the results of the full teacher model.

The development process of the student models was based on manual testing. Considering the small architecture, these methods were regarded as sufficient. The individual losses, seen in Table VI act as an indicator of how well each student replicates its assigned logits. The losses are considered to be small in comparison with the logit values, and thus the student models seem to fulfill their intended functionality.

The results indicate that the idea of training specialised students on some knowledge from the teacher may be a fruitful method of distilling a network. The student models were able to mimic some properties of the teacher models, with small losses, while occupying only 3 % of their memory. Considering the naîve choice of the method for assigning logits to the individual student models, there is surely room for optimization which could result in further loss reduction.

C. Network of Student Models

The network of student models should be compared with the teacher model, as well as the pruned model. The approach of ambiguously splitting the logits between the students seem to work well, although other possible splits may yield even better results. A minor loss in accuracy was recorded compared to the full teacher model. However, this drawback could be considered reasonable when put in relation to the gains in memory reduction and the possibility of distributing the teacher model. The same argument applies when comparing the network of student models with the pruned network.

Actual testing of parallel computing requires the models to be stored on different devices and executed on multiple cores. In this case, the algorithms were all executed on the same

core and although this is not in fact an actual implementation where parallel computing was tested, the model partitioning is still of highest relevance.

D. Future Work

It is generally hard to understand how a large neural network is processing data and to describe the flow of parameters through the layers. This makes a calculated split of the logits between the student models difficult to execute. Further research on the subject could result in a more intelligent split for maximum accuracy.

Another important research area is comparisons with other knowledge distillation and compression techniques for a more detailed evaluation of the method. It is possible that other methods of knowledge distillation/compression would have been more suitable in this case.

In addition, the procedure should be tested on more complex tasks and neural networks, which could yield different results. This includes multi classification problems as well as more complex data for binary classification problems.

VII. Conclusions

The pruned network managed to predict occupancy with a slightly higher mean accuracy than the network of students, although the difference can be considered insignificant. The network of student models, on the other hand, was smaller than the pruned model and had the advantage of enabling partitioning of the network and storage on multiple devices.

Even though the complete network of student models might not be able to fit on edge devices, the individual students might. Thus, the method of partitioned logit based knowledge distillation could be a solution to decentralizing calculations in IoT networks, or at least lead further development in the right direction.

Acknowledgment

The authors would like to thank KTH Live-In Lab for kindly giving us access to their data and associate professor Ragnar Thobaben for his invaluable guidance while supervising this project.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, Cambridge, MA, 2016, ch. 12, pp. 438–440. [Online]. Available: http://www.deeplearningbook.org

[2] S. Naveen and M. R. Kounte, "Key technologies and challenges in IoT edge computing," in *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2019, pp. 61–65.

[3] Y. Shi, L. Nguyen, S. Oh, X. Liu, F. Koushan, J. R. Jameson, and D. Kuzum, "Neuroinspired unsupervised learning and pruning with subquantum CBRAM arrays," *Nature Communications*, vol. 9, no. 1, p. 5312, Dec 2018.

[4] I. H. Shin, Y. H. Moon, and Y. J. Lee, "Towards understanding architectural effects on knowledge distillation," in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, 2020, pp. 1144–1146.

[5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, Cambridge, MA, 2016, ch. 6, p. 194. [Online]. Available: http://www.deeplearningbook.org

[6] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *CoRR*, vol. abs/1710.05941, Oct 2017. [Online]. Available: http://arxiv.org/abs/1710.05941

[7] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, June 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[8] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[9] A. Y. Ng, "Feature selection, L1 vs. L2 regularization, and rotational invariance," in *Proceedings of the Twenty-First International Conference on Machine Learning*, ser. ICML '04. New York, NY, USA: Association for Computing Machinery, 2004, p. 78. [Online]. Available: https://doi.org/10.1145/1015330.1015435

[10] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. 10, pp. 281–305, Feb 2012. [Online]. Available: http://jmlr.org/papers/v13/bergstra12a.html

[11] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'12. Red Hook, NY, USA: Curran Associates Inc., 2012, p. 2951–2959.

[12] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," in *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*, Jul 2010, pp. 1015–1022.

[13] M. Zhu and S. Gupta, "To prune, or not to prune: Exploring the efficacy of pruning for model compression," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net, 2018. [Online]. Available: https://openreview.net/forum?id=Sy1iIDkPM

[14] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural networks," *CoRR*, vol. abs/1506.02626, Oct 2015. [Online]. Available: http://arxiv.org/abs/1506.02626

[15] J. H. Wong and M. Gales, "Sequence student-teacher training of deep neural networks," *Interspeech 2016*, pp. 2751–2765, Sep 2016.

[16] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learning and Representation Learning Workshop*, 2015. [Online]. Available: http://arxiv.org/abs/1503.02531

Modelling of the DNA Helix's Duration for Genome Sequencing

Rim Sharif and Sabina Dzubur

***Abstract*—Nanopore sequencing is the next generation of sequencing methods which promises to deliver cheaper and more portable genome sequencing capabilities. A single DNA or RNA strand is passed through a nanopore nested in an artificial membrane with an electric potential applied across it. The nucleotide bases of the helix then interact with the ionic current in the nanopore, resulting in a unique signal that can be translated into the correct corresponding nucleotide sequence. This project investigated whether features of the raw signal data could be used as predictive indicators of the duration time of each nucleotide base in the nanopore. This is done in order to segment the signal before translation. The training data set used came from the sequenced DNA molecules of an *E. Coli* bacterium. Distribution candidates were fitted to a histogram of the duration data of the training set. Features of the current signal and distribution parameters were correlated in order investigate if a linear predictive model could be created. The results indicate that the feature zero-crossings is not an optimal option for construction of a linear model, while the large jumps and moving variance features often generate linear patterns. The μ parameter of the Log-logistic distribution had the best fit with the lowest relative root mean square deviation (rRMSD) of 2.7%.**

***Sammanfattning*—Nanopore sequencing är nästa generationens metod för DNA sekvensering som kommer att bidra med billigare och mer portabla sekvenseringsmöjligheter. Metoden innebär att en enkelsträngad DNA eller RNA molekyl passerar genom porer i nanostorlek, placerade i ett artificiellt membran samtidigt som en elektrisk potential appliceras över membranet. Nukleotiderna i genmolekylen interagerar med jonströmmen i poren, vilket resulterar i en unik signal som kan översättas till den korresponderande sekvensen av nukleotider som passerat. Detta projekt gick ut på att undersöka om egenskaper från signalen kan användas som predikativa indikatorer för varaktigheten som varje nukleotid befinner sig i membranporen. Detta för att sedan kunna segmentera signalen före översättningen till DNA sekvensen. Träningsdata som användes är sekvenserad DNA från en *E. Coli* bakterie. Kandiderande sannolikhetsfördelningar anpassades till ett histogram som beskriver varaktigheten. Egenskaperna och parametrar från fördelningarna korrelerades för att skapa en linjär modell. Resultatet visade att antalet skärningar i x-axeln som signalegenskap inte är det optimala valet för konstruktion av en linjär modell. Skillnaden mellan två signalvärden som är mindre än en varierbar konstant och glidande variansen som signalegenskaper genererar ofta linjära mönster. Resultatet visade även att sannolikhetsfördelningen Log-logistic hade lägst relativ medelkvadratavvikelse (rRMSD) på 2.7%.**

***Index Terms*—Nanopore sequencing, DNA sequencing, Linear regression, Supervised learning, Distribution fit.**

Supervisors: Joakim Jaldén and Xuechun Xu

TRITA number: TRITA-EECS-EX-2021:187

I. Introduction

Genome sequencing allows researchers to determine the genetic structure of biomolecules by segmenting a DNA or RNA molecule into its nucleotide bases. The outcome provides the ability to detect diseases and to choose the right personalized treatment. Compared to traditional sequencing methods, there are several advantages to Nanopore DNA sequencing. Nanopore sequencing offers real time reads, and is cheaper than current short-read platforms. Moreover, it can sequence DNA molecules of any length. This is an important feature because long reads provide vital information on how distal sequences are spatially related and helps reveal the full spectrum of genome variation [1]. However there are improvements needed for nanopore sequencing before it can be reliably used, since the method still has a lower accuracy than traditional methods [2].

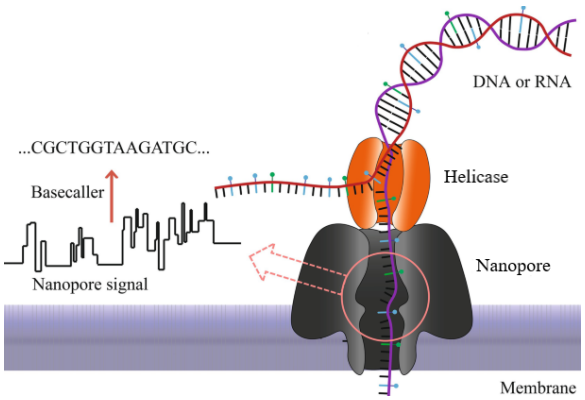


Fig. 1. A single stranded DNA sequenced by passing through a helicase placed atop the nanopore. Figure taken and modified from [5].

Since Nanopore sequencing is being developed with the aim of reducing costs and time, DNA sequencing will become more accessible. In this way Nanopore sequencing will contribute to safer performance of medical procedures in many fields, such as prenatal diagnosis when performing an amniocentesis. Currently this procedure is guided by an ultrasound that determines the position of the fetus, and a needle is inserted through the abdominal wall into the uterus to withdraw a small amount of amniotic fluid. The fluid surrounding the fetus contains DNA from which valuable information can be collected about the baby's health, but the procedure carries various risks such as miscarriage [3]. The accessibility of Nanopore sequencing could make it possible to perform non-invasive prenatal testing of the fetus through a blood-sample from the carrying parent which does not imply any risk to the fetus [4].

The concept of Nanopore sequencing, see Fig. 1, relies on reading ionic current data generated from the nucleotide bases of a single stranded DNA molecule passing through

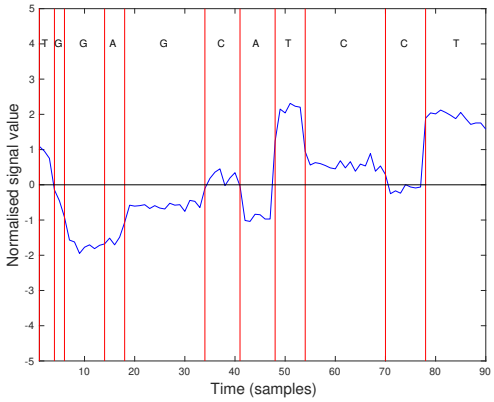


Fig. 2. Normalized current signal of an *E. Coli* DNA molecule that has been sequenced, segmented and labeled with the corresponding nucleotides.

a nano-scale hole (a nanopore) of an electrical measurement device [6]. The device is built upon a nanopore embedded in an artificial membrane that separates two regions, when an electrical potential is applied over the membrane it results in an electrical current and flow of nucleotides. As the DNA strand moves through the nanopore, the physical properties of the nucleotide bases, such as size and chemical structure, will cause blockages that are modulated as variations of the electric current which is sampled at a fixed frequency [7].

The received raw electric current signal is translated into a DNA sequence by identifying segments of the signal that corresponds to specific nucleotides as shown in Fig. 2. However, during translocation through the nanopore, sequencing errors can occur and result in a low signal-to-noise ratio [2]. Several factors contribute to sequencing errors, such as the structural similarity of the nucleotide bases, simultaneous influence of multiple bases on the signal, and the fact that the signal does not change within homopolymers, sequences of the same nucleotide succeeding one another. This requires a sophisticated base-calling software to be developed in order to process the generated data. Base-calling is the term used to describe the process of translating the current signal into base sequences [2], [8].

Unconstrained translocation through the nanopore is too fast to record [2]. Therefore, a DNA helicase molecule is placed on top of the pore to pass the DNA through, one base at the time. DNA helicases are enzymes that use energy from hydrolysis of adenosine triphosphate (ATP) for this task. It is also the helicase that is responsible for unwinding the double stranded DNA molecule to a single stranded one [9]. There are two main complications caused by the helicase that result in an inconsistent duration time for the bases. The first is that the helicase pulls the DNA strand at a nonuniform speed, and the second is that the concentration of ATP decreases along the experiment [2]. However, a model of the duration time of the bases in the pore could be used to solve these issues.

Previous base-callers have utilized different models and algorithms for different stages of the base-calling process. These range from ones based on the hidden Markov model (HMM), to various neural networks such as recurrent neural networks (RNN) and convolutional neural networks (CNN) [2], [8]. DeepNano is an example of an independently developed base-caller that translates the signal into events and uses a bidirectional RNN to predict the base sequences by modelling statistical characterizations of the events. Chiron is another independently developed base-caller that combines a CNN and an RNN. The CNN is used for detecting patterns in the data without first translating it into events, while the RNN is used to predict polymer probabilities [2], [8].

The purpose of this project is on a smaller scale and involves the first steps of base-calling. The aim is to determine if it is possible to model and parameterize the duration data, which is in this case defined as the number of samples taken per nucleotide base that is in the nanopore. This in order to contribute to the segmenting process of base-calling. By using sequenced data from an *E. coli* bacterium as the training data set, this project investigated if a linear model could be established with certain features of the raw signal.

II. Method

Histograms were produced for different reads for visualizing the probability of a base remaining in the pore for a certain number of samples. The number of samples takes per base can vary between 1 sample and over 250 samples. However, it is relatively rare that a base remain in the nanopore for much longer than even 60 samples, see Fig 3 for a typical histogram. Therefore, when modeling this experiment it is appropriate to disregard bases with abnormally long sampling time. In this project all bases with a duration time of over 60 samples were truncated.

The distribution of the number of samples taken per base in a given read tends to be heavily right skewed with a long tail, see Fig. 3. As such, it was empirically decided to investigate six different probability distributions as potential candidates for modeling the duration data. The relationship between the signal features and the fitted distribution's parameters was then analyzed using linear regression.

The training data set used in this project consisted of 10,279 read segments collected at different phases of the sequencing experiment and stored in three data files. Long DNA sequencing results in several gigabytes of data, and each read segment contains the sequenced data for several thousands of DNA nucleotide bases. Therefore, the best compromise for this project in terms of computational time, and quality of results, was to iterate experiments on a portion of the total data points. The chosen portion was divided up so that it contained reads evenly spread across all 10,279 read segments. Each feature was compared to each distribution parameter for a total of 900 read segments. These data points were then plotted to see if a pattern could be observed. If a linear model could be established, the corresponding parameter is categorized as a ”line” parameter, and if not, then as a ”cloud” parameter. All experiments and plots were produced using Matlab. Furthermore, the current signal was normalized prior to any of the experiments for easier plotting and referencing purposes.

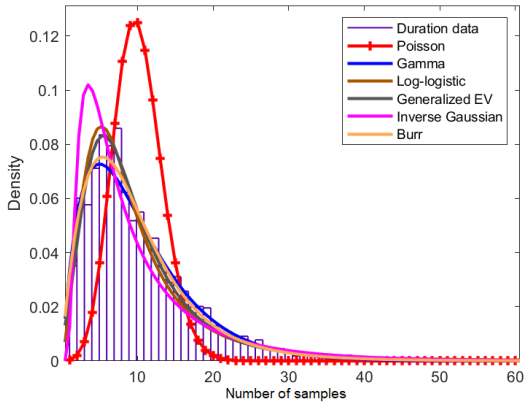


Fig. 3. Typical probability distribution for the duration data in a given read.

TABLE I
LIST OF THE DISTRIBUTIONS INVESTIGATED

Distribution	Parameters	Parameter description
Poisson	λ	Mean
Gamma	α	Shape parameter
	β	Scale parameter
Log-logistic	μ	Mean
	σ	Scale parameter
Inverse Gaussian	μ	Scale parameter
	λ	Shape parameter
Burr Type XII	α	Scale parameter
	c	Shape parameter
	k	Shape parameter
Generalized Extreme Value	k	Shape parameter
	σ	Scale parameter
	μ	Location parameter

A. The distributions

Using Matlab, a variety of probability distributions were fitted to a histogram of the duration data of the training set. There are 22 options available with Matlab's distribution fitter `fitdist` (listed in Appendix A), all of which were compared in relation to the histogram. Parameter values for the fitted distributions were estimated using maximum likelihood estimation with a 95% confidence level [10], and then compared graphically in a plot overlaying a histogram of the duration data. Out of the 22 compared options, six distribution candidates, shown in Fig. 3, were selected that were deemed to have the most favorable fit: Poisson, Gamma, Log-logistic, Inverse Gaussian, Generalized Extreme Value, and the Burr Type XII distribution. The equations for the probability distribution functions are presented in Appendix A. As seen in Fig. 3, the Poisson distribution differs from the other distributions, but it was still selected because of its one parameter dependence. The distributions come from one to three parameter-family distributions, see Table I for further details.

B. Features

Zero-crossings and large jumps were the main attributes used for identifying features of the raw signal data. These were chosen as they seem to be somewhat indicative of a base change in the signal. Zero-crossings is simply defined as the average number of samples per zero-crossing, i.e. the average number of times the signal crosses the x-axis in a given read segment, see Fig. 2 for reference. The large jump feature defines a large jump as,

$$|y_i - y_{i+1}| > \delta \tag{1}$$

where y_i is the signal value of the ith sample, and δ is the threshold that defines a large jump. As such multiple features were analyzed by varying the threshold variable. The current signal has some noise, and was therefore also modified with a mean or median sliding window before analysis.

The third feature investigated was the mean variance of a read segment within a sliding window. This was also chosen since the signal variance theoretically could reflect the change of nucleotide bases. Matlab's function `movvar` was used to compute the variance as,

$$V = \frac{1}{1-N}\sum_{i=1}^{N}|y_i - \mu|^2 \tag{2}$$

where y_i is the signal value, μ is the mean signal value within a window of size N [11]. Experiments were computed with different window sizes since that is an important factor for this feature. If N is set to an odd value the window is centered at the current position of the element in the signal vector. If N is even, the window is centered at the current and previous element. The variance is calculated for elements that fill the window, if there are not enough elements at the signal's endpoints, they are truncated.

Both the large jump and moving variance features will produce different result depending on the different window sizes used. Throughout the experiments different window sizes are of more, or less, interest depending on the feature and type of window. Therefore, they were varied differently for each experiments, see section III for more details.

C. Parameter estimation

The underlying relationship between the parameter of the fitted distribution and the feature can potentially be described with the regression model,

$$y_n = \alpha + \beta x_n + \varepsilon_n \tag{3}$$

where y_n is the parameter value of the nth training example, x_n is the calculated feature, ε_n is the error deviation, β and α are the slope and intercept respectively [12].

The least-squares method was used to fit the regression line and minimize the error deviation with the minimization problem,

$$\min_{\alpha,\beta} \quad \sum_{n=1}^{N}(y_n - \alpha - \beta x_n)^2 \tag{4}$$

where N is the number of total training examples. Least-square line fitting could also be done in Matlab using the `polyfit` function [13].

In order to measure the validity of the fit, the relative root mean square deviation (rRMSD) of the feature was used to

TABLE II
ZERO-CROSSING AND DISTRIBUTION PARAMETER RELATIONSHIP

Distribution	Parameter	Cloud/Line	rRMSD
Poisson	λ	Cloud	0.117
Gamma	α	Cloud	0.078
	β	Cloud	0.176
Log-logistic	μ	Cloud	0.051
	σ	Cloud	0.055
Inverse Gaussian	μ	Cloud	0.117
	λ	Cloud	0.118
Generalized Extreme Value	k	Cloud	0.136
	σ	Cloud	0.144
	μ	Cloud	0.118

assess the quality of the regression model. The RMSD is calculated as,

$$RMSD = \sqrt{\frac{\sum_{n=1}^{N}(y_n - \alpha - \beta x_n)^2}{N}} \tag{5}$$

and the rRMSD is defined as,

$$rRMSD = \frac{RMSD}{\bar{y}} \tag{6}$$

where the RMSD is divided by the expected value of the distribution parameter.

III. RESULTS

The best results for each parameter correlation are shown in Table II to Table VI. However, Matlab failed to fit the Burr Type XII distribution to all of the reads during the experiment which is why no results were achieved. The shape parameter k of the distribution diverged to infinity for some of the reads, leading to a crash in the Matlab program. Consequently, the Burr Type XII distribution was eliminated as a candidate.

Table II shows the results of how the zero-crossing feature is related to different distribution parameter values. All of the parameters resulted in a cloud-like scatter plot similar to the results in Fig. 4 when plotted with the zero-crossing feature. The data points in Fig 4 and Fig. 5 are displayed in three different colors, wherein each one was extracted from its corresponding file and represents a part of the signal collected at a different time in the sequencing experiment. Red data points originate from around the start of the sequencing, black from the middle, and blue from the later parts of the signal.

Table III shows the results of the large jump feature when derived from an unmodified current signal. The feature was tested with the jump threshold δ ranging from 0.3 to 1 for each distribution parameter. All results outside this range did not yield any desirable results and were not investigated further. As the results show in Table III, six parameter-feature correlations did provide a linear fit, while the rest gave a cloud-like plot. However, the μ parameter of the Log-logistic distribution performed the best with an rRMSD of 3% when the jump threshold was set to $\delta = 0.7$.

Table IV shows the results of the large jump feature when derived from a signal that has been modified with a

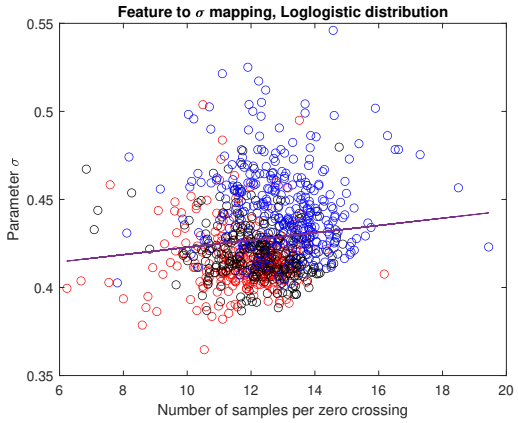


Fig. 4. Cloud-like relationship of the zero-crossing feature and the σ parameter in a Log-logistic distribution fit. Each color represents a different stage of the sequenced data.

TABLE III
UNFILTERED SIGNAL: LARGE JUMP AND DISTRIBUTION PARAMETER RELATIONSHIP

Distribution	Parameter	Threshold	Cloud/ Line	rRMSD
Poisson	λ	0.8	Line	0.062
Gamma	α	0.7	Cloud	0.060
	β		Line	0.083
Log-logistic	μ	0.7	Line	0.030
	σ		Cloud	0.051
Inverse Gaussian	μ	0.8	Line	0.062
	λ		Cloud	0.118
Generalized Extreme Value	k	0.7	Cloud	0.132
	σ		Line	0.081
	μ		Line	0.067

sliding averaging window. The feature was plotted against each distribution parameter with all possible combinations of threshold δ and window size. In order to find the best correlation conditions, the δ variable ranged from 0.3 to 0.7, and the window size was varied between three and five. The results obtained show that in all cases a threshold value of 0.3, and a window size of three samples had the best outcomes.

Six linear relationships could be established for the different distribution parameters. One distribution had the lowest parameter-feature rRMSD, which is the μ parameter for the Log-logistic that gave a deviation of 2.7%.

Table V shows the best results achieved with the large jump feature when the signal has been modified with a sliding median window. As before, the feature was plotted with each distribution parameter with all possible combinations of δ and window sizes. However, since a median filter does not smooth out a signal as much as a mean one, the window sizes varied between three and eight. Similar to the results of a sliding mean window, Table V shows eight linear relationships, wherein the best line fit with the lowest rRMSD is also the μ parameter of the Log-logistic distribution. This result was specifically achieved with a jump threshold of 0.5 and a window size of three samples, giving a deviation of 2.7%,

TABLE IV
SIGNAL PROCESSED WITH A MEAN FILTER: LARGE JUMP AND DISTRIBUTION PARAMETER RELATIONSHIP

Distribution	Parameter	Threshold	Window	Cloud/ Line	rRMSD
Poisson	λ	0.3	3	Line	0.050
Gamma	α β	0.3	3	Cloud Line	0.059 0.072
Log-logistic	μ σ	0.3	3	Line Cloud	0.027 0.048
Inverse Gaussian	μ λ	0.3	3	Line Cloud	0.050 0.121
Generalized Extreme Value	k σ μ	0.3	3	Cloud Line Line	0.132 0.061 0.059

TABLE V
SIGNAL PROCESSED WITH A MEDIAN FILTER: LARGE JUMP AND DISTRIBUTION PARAMETER RELATIONSHIP

Distribution	Parameter	Threshold	Window	Cloud/ Line	rRMSD
Poisson	λ	0.3	4	Line	0.050
Gamma	α β	0.3	6	Cloud Line	0.058 0.069
Log-logistic	μ σ	0.5	3	Line Cloud	0.027 0.048
Inverse Gaussian	μ λ	0.5	3	Line Cloud	0.049 0.120
Generalized Extreme value	k σ μ	0.3	4	Cloud Line Line	0.132 0.060 0.061
Generalized Extreme Value	k σ μ	0.3	3	Cloud Line Line	0.132 0.061 0.059

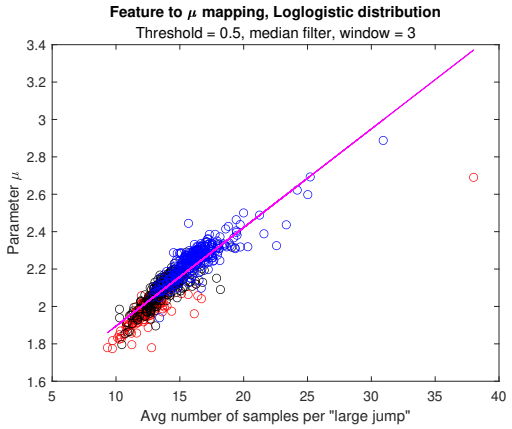


Fig. 5. Linear relationship of the large jump feature and the μ parameter in a Log-logistic distribution fit. Each color represents a different stage of the sequenced experiment.

as shown in Fig. 5.

Table VI shows the best results achieved for the moving variance feature where six linear relationships could be established. Experiments were computed with a sliding window

TABLE VI
UNFILTERED SIGNAL: MOVING VARIANCE AND DISTRIBUTION PARAMETER RELATIONSHIP

Distribution	Parameter	Window	Cloud/ Line	rRMSD
Poisson	λ	19	Line	0.064
Gamma	α β	20	Cloud Line	0.067 0.097
Log-logistic	μ σ	18	Line Cloud	0.031 0.046
Inverse Gaussian	μ λ	18	Line Cloud	0.031 0.046
Generalized Extreme value	k σ μ	20	Cloud Line Line	0.135 0.076 0.075

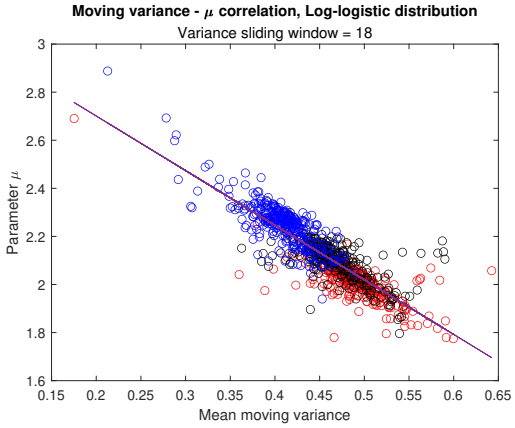


Fig. 6. Linear relationship of the moving variance feature and the μ parameter in a Log-logistic distribution fit. Each color represents a different stage of the sequenced experiment.

that ranged from two to twenty-two in order to find the lowest rRMSD for the linear correlations. The results show that the best correlations were achieved with both the Log-logistic and the Inverse Gaussian distributions. The mean parameter μ and the scale parameter μ both had a deviation of 3.1% when the variance window was 18 samples long, see Fig. 6.

Since the Log-logistic distribution had the lowest deviation when the mean parameter μ was correlated with the feature large jumps, the linear equation for the parameter is presented. The equation was calculated as,

$$\mu_{LL} = 1.3678 + 0.0527x_n \tag{7}$$

where x_n is the average number of samples per jump larger than 0.5, and the normalized signal has been modified with a mean or median sliding window of three samples.

IV. DISCUSSION

The main goal of this project was to investigate if it is possible to model and parameterize the duration data, i.e. the number of samples per nucleotide base. This is in order to create a linear model based on the correlation of distribution parameters and features of the raw current signal. The experiments show that it is possible to establish a linear model

for certain parameters of five specific distributions using three signal features. Interpretation of the results will be discussed in this section.

A. Correlation of parameters

From the computations of the signal feature and the distribution parameter for each read, the obtained results were presented as color coded plots. As previously mentioned, each given data file contained reads from different occasions of the nanopore sequencing, and by separating the occasions by color some insights could be gathered. All plots produced of the mean parameters shared the same structure in which parameter and feature value steadily increase from the earlier parts of the read to the later ones. This means that the number of large jumps in a read segment decreases with time since there are more samples in between them. This is clearly reflected in Fig. 5, and is an expected result as it indicates that the duration time of a base in the nanopore increases as times goes on. The reason for this is because the phenomena depends on the helicase enzyme placed on top of the nanopore during the sequencing. The enzyme uses energy from ATP hydrolysis to pass the nucleotides through the nanopore. This causes the ATP concentration to decrease during the process, which in turn leads to slower translocation of the DNA strand [2].

The obtained plots generally resulted in two main pattern structures, either a cloud-like pattern or a linearly fitted one. A cloud structure indicates that there is no linear correlation between the signal feature and fitted distribution parameter, while the latter is a clear indication that a linear correlation exists. Figure 4 and Figure 5 are examples of the two patterns. It is possible that definition of a distribution parameter and a feature do not match, and therefore the feature could perform weakly when correlated. It is also difficult to draw the absolute conclusion whether a feature is unsuitable from only one plot. Therefore, several computations with different distributions and signal modifications are necessary to determine if the feature is effective based on the generated pattern.

Table II clearly shows that the zero-crossing feature is ineffective as all parameter-feature plots resulted in a very cloud-like structure, see Fig. 4 for an example. On the other hand, Tables IV, V, and VI show that a linear model could be established for at least one of the parameters of all the distribution candidates. This indicates that the large jump and the moving variance features can be used to estimate certain parameters that are related to the mean, scale, or location of a distribution, see section III for details on which parameters.

Table IV of the large jump feature shows that the optimal window size of the mean filtering window was three samples for all five distributions. This is not strange as an averaging filter will smooth out the current signal, and a larger window size would have a greater effect on the signal. In the case of a feature that is based on large jumps, the filter could potentially counteract the signal variation.

The results of the moving variance feature show that the best correlations were achieved with a window size of 18 to 20 samples, see Table VI. This is a relatively large window since the average duration for a base is between six and eight samples, see Fig. 3. One possible explanation for this is that a window of 18 samples roughly corresponds to the duration time for two nucleotide bases. In this sense it is possible that the moving variance feature is related to the large jump feature, which could explain why they perform similarly.

Tables III, IV, V, and VI show that the shape parameters of the distributions performed the worst with both the large jump and moving variance features. This is probably because shape parameters represent the skewness and kurtosis of a distribution. Since both features act as some sort of indicator of the average number of bases in a read, it will most likely correspond to parameters that are related to the mean of the distribution. This insight implies that features that correspond to the shape parameters in some way are needed in order to provide a good estimate for them.

B. Deviation of model

The performance of the created linear models were measured with the relative root mean square deviation. An important factor of the rRMSD in this project is that it is only effective as a measurement of fit quality if a linear pattern could be created. This is especially apparent in Fig. 4 where the rRMSD is 5.5%, however it is clear from the image that there is no real correlation between the feature and fitted σ parameter.

Depending on accuracy requirements it is possible that a feature could still be used to estimate a parameter despite a cloud-like correlation. Fig. 4 of the zero-crossing feature shows that the σ parameter of the Log-logistic distribution varies relatively little in value. This means that depending on the implementations of future projects, it could still be possible to use the zero-crossings feature, or have it as a set constant to estimate the parameter.

C. Further research

In future research projects within the field of Nanopore DNA sequencing, investigation of more complex distributions from several parameter-families could be done since this project applied simple distributions to the linear model. The distribution fitting might use other or several estimation methods and not only be selected from visual inspection but also motivated mathematically. Additional statistical characterizations of the raw signal data are suggested for further feature investigations. Also, the variance feature could be investigated further with the signal modified with a mean or median filter. Further careful investigation of distributions and features contributes to the analysis of whether they yield more accurate results when implemented with a regression estimator.

This project had complications with a long execution time when investigating each feature with all the reads. The implementation restricted the number of data points used in the result plots to a relatively low number compared to the number of data points in a read. Since it is desirable to use as many data points as possible further studies could be to optimize the implementation and strive towards a more time effective one.

Further research can be done within neural networks as well, and the investigation of their effectiveness in boundary

prediction for segmenting the raw current signal. Also, more advanced methods for distinguishing homopolymers might be of interest and another compartment of the Nanopore DNA sequencing field.

V. Conclusion

The capability of features acting as predicative indicators of the duration time of each nucleotide base in the nanopore varies. In this case the feature zero crossing correlated with any other distribution parameter results in a cloud, which indicates the feature to be ineffective. On the other hand, the features large jump and moving variance correlated with distribution shape parameters appears as clouds, but the features often generated linear patterns when correlated with other distribution parameters. Specifically the features act as good predictive indicators and the rRMSD reaches its lowest values when used with the Log-logistic and Inverse Gaussian distributions. This concludes that these two features and probability distributions are the most promising options for establishing a linear model.

Appendix A
Matlab probability distributions

Acknowledgment

The authors would like to thank supervisors Joakim Jaldén and Xuechun Xu for their invaluable patience, support, and encouragement during the project process.

References

[1] G. A. Logsdon, M. R. Vollger, and E. E. Eichler, "Long-read human genome sequencing and its applications," *Nature Reviews Genetics*, vol. 21, no. 10, pp. 597–614, Oct 2020. [Online]. Available: https://doi.org/10.1038/s41576-020-0236-x

[2] F. Rang, W. Kloosterman, and J. De Ridder, "From squiggle to basepair: computational approaches for improving nanopore sequencing read accuracy," *Genome Biology*, vol. 19, Jul. 2018.

[3] (2020, Nov) Amniocentesis. [Online]. Available: https://www.mayoclinic.org/tests-procedures/amniocentesis/about/pac-20392914

[4] S. H. Cheng, P. Jiang, K. Sun, Y. K. Y. Cheng, K. C. A. Chan, T. Y. Leung, R. W. K. Chiu, and Y. M. D. Lo, "Noninvasive Prenatal Testing by Nanopore Sequencing of Maternal Plasma DNA: Feasibility Assessment," *Clinical Chemistry*, vol. 61, no. 10, pp. 1305–1306, 2015. [Online]. Available: https://doi.org/10.1373/clinchem.2015.245076

[5] M. He, X. Chi, and J. Ren, *Applications of Oxford Nanopore Sequencing in Schizosaccharomyces pombe*. New York, NY: Springer US, 2021, pp. 97–116.

[6] (2020, Jun) How it works. [Online]. Available: https://nanoporetech.com/how-it-works

[7] Y. Wang, Q. Yang, and Z. Wang, "The evolution of nanopore sequencing," *Frontiers in Genetics*, vol. 5, p. 449, 2015. [Online]. Available: https://www.frontiersin.org/article/10.3389/fgene.2014.00449

[8] J. Zeng, H. Cai, H. Peng, H. Wang, Y. Zhang, and T. Akutsu, "Causalcall: Nanopore basecalling using a temporal convolutional network," *Frontiers in Genetics*, vol. 10, p. 1332, 2020. [Online]. Available: https://www.frontiersin.org/article/10.3389/fgene.2019.01332

[9] J. M. Craig, A. H. Laszlo, H. Brinkerhoff, I. M. Derrington, M. T. Noakes, I. C. Nova, B. I. Tickman, K. Doering, N. F. de Leeuw, and J. H. Gundlach, "Revealing dynamics of helicase translocation on single-stranded dna using high-resolution nanopore tweezers," *Proceedings of the National Academy of Sciences*, vol. 114, no. 45, pp. 11 932–11 937, 2017. [Online]. Available: https://www.pnas.org/content/114/45/11932

[10] MathWorks. (2021, May) Fit probability distribution object to data - matlab fitdist. [Online]. Available: https://se.mathworks.com/help/stats/fitdist.html

[11] ——. (2021, May) Moving variance - matlab movvar. [Online]. Available: https://se.mathworks.com/help/matlab/ref/movvar.html

[12] G. Blom, J. Enger, G. Englund, J. Grandell, and L. Holst, *Sannolikhetsteori och statistikteori med tillämpningar*, 5th ed. Lund, Sweden: Studentlitteratur, 2005.

[13] MathWorks. (2021, May) Polynomial curve fitting - matlab polyfit. [Online]. Available: https://se.mathworks.com/help/matlab/ref/polyfit.html

CONTEXT M – PART II
INFORMATION ENGINEERING: BIG DATA & AI

POPULAR DESCRIPTION

Masks don’t make you anonymous!

Imagine walking through town on your way to meet your secret date. Your mask covering your face. Nobody must know your intentions. But the cameras are everywhere and you cannot hide! Your significant other receives a message, because the cameras have registered your face despite your hiding attempts. You’re screwed, all because of face recognition!

Recent events in Hong Kong and the ongoing pandemic has contributed to technological advancements in biometric identification. Extensive protests in the state have led to the police of Hong Kong being granted extended access to surveillance. This means having greater power to identify the protestors, and even tracking them down. With the pandemic in full swing, most protestors are using facial masks. Both to protect themselves from Covid-19, but also to mask their identity. Unwilling to accept anonymity, the Chinese government has developed facial recognition software that can identify people, from nothing but their eyes. Scary, isn’t it?

Previously, security cameras could not by themselves keep track of a large population of individuals. The images had to be analyzed by humans, making it very time consuming, and prone to error. With new face recognition software, algorithms can be trained to recognize a person from an image, or a video feed. By analyzing images of a certain individual, the algorithms find patterns in the facial features, from which the individual can be identified in other contexts as well. This allows for mass surveillance.

There are of course upsides to this incredible technology. Crime rates could be drastically reduced, and many offenders would be caught and locked up. However, the downsides to widespread use of face recognition software likely outweighs the positives, at least for now.

To summarise, masks might be good for Covid-19 but they won’t hide your identity!

SUMMARY OF PROJECT RESULTS

Big data and AI has become a large part of today's digital society. Everything, from personal advertisement to weather forecasts, uses AI for different reasons such as optimizing profit or making predictions more accurate. Extraction of useful information from data requires advanced machine learning. This is, for example, the case with image recognition. By analyzing images and training machine learning (ML) algorithms, such as deep neural networks, one can teach computers to identify people just by an image of their face, predict the weather from satellite pictures of the earth, or decode handwritten documents, to name a few examples.

Image recognition is studied by the project groups M5, M6 and M7. The project group M5 studied the effects of using facial recognition to distinguish between dolls and humans in photos. The M6 project addressed text segmentation in an effort to differentiate texts on old documents. In project M7 the group applied multi-staged neural networks and random forest to identify users based on fingerprints.

The M5 project group is looking into the beginning of facial recognition using Neural Networks. Neural networks are designed to be artificial versions of the human brain with the same kind of structure. This project contains the first step towards being able to distinguish between a picture of a human and pictures of dolls, and other objects.

The inability to distinguish between dolls and humans in photos have been seen in young children. The neural network acts in a similar fashion as the human brain, therefore the system should have the same problem if the system is insufficient. The next step is to test this against the performance of real children but in order to do so the system should be trained with a much larger dataset.

The M6 project group has identified and implemented several algorithms from scientific literature relating to the problem of blind source separation (BSS). When relating to images, this is often called the problem of show-through cancellation.

The phenomena can be observed when looking at old manuscripts where the text from the reverse side shows through to the front, or vice versa. The mathematical models that describe the show-through effect are nonlinear which means that extracting the original images from the mixtures is not trivial. The best results were obtained starting from a physical model, derived by Gaurav Sharma in 2001. The parameters of the model were estimated by maximising the Likelihood function of the model parameters, with respect to the scanned documents using a gradient ascent algorithm. Satisfactory results were also derived using Independent Component Analysis (ICA), a method for separating a multivariate signal into independent additive subcomponents. A segmentation method using information from both sides of the paper also yielded acceptable results. Future projects would do well to investigate the use of neural networks in solving similar nonlinear BSS problems, since they are known for being good nonlinear processing algorithms.

In project M7, machine learning based image recognition has been studied. The main goal has been to create a multi stage identification algorithm to identify 600 different users from the SOCOF-dataset, a set of 6000 fingerprints, one fingerprint per finger. Using information on gender, which finger and which hand the fingerprint is from, the dataset is split into smaller sets on which different instances of the supervised learning ML-model are trained. It is therefore necessary that the first instances of the ML model can distinguish which of these categories each fingerprint belongs to. The image recognition is divided into a basic single stage identification, two stage classification and N-stage method, showing varying results. Both deep neural networks and random forest have been investigated to create an image recognition software. Considerable time of the project was spent finding proper ways to expand the dataset. Further research could investigate different datasets and aim to find better ways of extending the dataset.

IMPACT ON SOCIETY AND ENVIRONMENT

Big data, machine learning and, more specifically, image recognition are growing more common in every part of society by the day. In health care, images are often used for diagnosis. It can be anything from large x-rays and ultrasound images, to a tiny image of a few cells. As of today, most of these images are analyzed by physicians, and as is the case for all humans, physicians do make mistakes. By using machine learning to analyze images, the number of mistakes can be decreased. Not only will the algorithms be less ambiguous in their choice of diagnosis in uncertain cases, but each new case could also help improve the algorithms for every care facility that uses them. In comparison to spreading every new piece of information to every physician who would benefit from receiving it, this could potentially increase the rate of development of health care significantly.

Data analysis can be used to perform surveillance and facial recognition. Surveillance can, in turn, be used to prevent crime, or solve crimes already committed. Surveillance can also be used against individuals to make sure they comply with rules and regulations. The technique is used in Hong Kong to make sure people who are supposed to be in Covid-19 quarantine, actually stay quarantined. Another example comes from China, which uses data from social media to track down individuals who voice politically sensitive opinions.

As the use of the Internet has increased in society, the need for storage of large amounts of data has followed the same trend, resulting in larger and larger data centers. Data centers are extensive complexes where data servers are run to store and process large amounts of data.

These buildings are spread out all over the world to help industries and individual users in their daily life. The impact on the environment from the data centers has steadily increased over the years, requiring substantial amounts of energy to both run and cool the systems. To put it into perspective, a single server hall can require as much energy as a medium size town.

This means that the complex could require its own power plant just to keep it running. An interesting conclusion from this is that when someone uses a search engine, one search equals boiling three liters of water.

A majority of the energy that is consumed by the computers in the server hall is converted to heat. To counteract the effect of overheating the machines, cooling systems are implemented. These systems also require energy, which further impacts the environment. The heat from the servers has to go somewhere, which usually involves the local surroundings. Some servers utilize seawater for cooling, whilst other companies have placed their stations in naturally cold areas, reducing the need for additional cooling. The excess heat generated by the halls can have a substantial impact on the local environment. Sea life, plant life and wildlife are all affected. Another approach, which has seen increased interest in recent years, is to transfer the excess heat to local apartments and residents, thereby minimising the energy usage of the public, while contributing to the technological advancement of society.

Large amounts of personal biometric data can be used for many things. For example, a fingerprint can be used as a password to your computer or mobile phone, and large amounts of image records could be used to diagnose skin cancer. These modern wonders have a positive impact on society, simplifying life and freeing time, but it comes at a price. Even though biometric data in itself is neither good nor bad, one has to be careful with how it is used. For example, every fingerprint is unique and does not change during a person's lifetime. This means that it can be used to identify an individual, which is useful for both unlocking phones and identifying criminals. Because of the ever growing digitalization, more people are beginning to use their fingerprints as a password for different applications, and as such, more and more fingerprints are available in an increasing number of databases. Because of the extensive use of fingerprints as security locks, the theft of a fingerprint could give an unauthorized person access to a wide variety of secure domains, such as personal computers. Furthermore, law enforcement does, for example, also use fingerprints to identify criminals, and hence, if someone has access to a set of fingerprints they could be used in criminal activity.

In the healthcare system, there are many regulations regarding storage, and use, of sensitive personal information. A majority of the Swedish population trust that the hospitals abide by these rules and therefore trust them with their information. Most people have faith that the hospitals abide by these rules and therefore trust them with their information. Regarding companies, it is unknown how personal data will be handled and how secure their systems are. Hence, it is hard to assess how great of a risk it is that the information gets stolen and leaked. If that were to happen, the information could, for example, be used by an insurance company to exploit the knowledge of personal health, based on DNA. If one was prone to have a certain disease that knowledge may be used to increase insurance costs for that person.

The context group cannot determine if collecting personal data is right or wrong. It can help save lives and make life easier, but in the wrong hands it could also do a lot of damage. We would recommend everyone to think twice before providing such sensitive information. In addition, laws are not always benign. Using data to assure absolute compliance is therefore problematic. Another problem is that if infrastructure exists that allows surveillance, there is a risk that it can be used by malign forces. Furthermore, the context group is of the opinion that freedom of speech should be absolute. Usage of data analysis to track down individuals who exercise this right is misuse.

A Small Classification Experiment Between Dolls and Humans With CNN

Ylva Reinders and Josefin Runnstrand

***Abstract*—This study is about a small experiment using CNN models to see how well they differentiate between dolls and humans. The experiment used two different kinds of CNN models one which was built after a classic model and one more rudimental model. The models were tested on how accurately they predicted the right answer. The experiment was a three-classed problem and had a set of different parameters to test what would make it harder for the system to classify the images correctly. The original images were digitally enhanced to test different conditions. The models were tested on a dataset with negative images of the original images, one set with higher contrast than the original, one set with different light conditions, one set with higher brightness and three different levels of low resolution on the images. The study concludes that brightness and lighting are the two most difficult conditions. The contours in the image are the most important part for successful classification.**

***Sammanfattning*—Studien är på ett litet experiment med CNN-modeller för att se hur väl de skiljer mellan dockor och människor. Experimentet använder två olika typer av CNN-modeller, en som byggdes efter en klassisk modell och en mer rudimentär modell. Modellerna testades på hur exakt de kan bestämma de olika klasserna. Experimentet var ett treklass problem och bilderna testades med olika typer av förhållanden, för att se vad som skulle göra det svårare för modellen att klassificera bilderna korrekt. Original bilderna gjordes om för att studera olika typer av förhållanden. Modellerna testades på ett dataset med negativa bilder av originalbilderna, en uppsättning med högre kontrast än originalet, en uppsättning med olika ljusförhållanden, en uppsättning med högre ljusstyrka och tre olika nivåer med låg upplösning av bilderna. I studien drogs slutsatsen att ljusstyrka och belysning är de två svåraste förhållandena. Konturerna på objekten i bilden är den viktigaste faktorn för en framgångsrik klassificering.**

***Index Terms*—CNN, image recognition, classification, experiment.**

Supervisors: Saikat Chatterjee

TRITA number: TRITA-EECS-EX-2021:188

I. Introduction

In today's society image and facial recognition systems are widely used. Facebook uses them to recognize people in pictures [1] and it can also be used in identification when you log in on your computer or mobile phone. Facial recognition is also a hot topic in security and crime-fighting [2]. Therefore the facial recognition systems must be as accurate as possible. It is harder to identify a face than an eye or a fingerprint. The face isn't constant and will change throughout a person's lifetime, a person's features can also change because of outside interference like cosmetics and plastic surgery [2]. A facial recognition system can also struggle depending on the pose and angle of the subject. So by identifying conditions that make it harder for the systems to identify correctly actions can be taken so that the systems will be used more efficiently.

Faces aren't the only thing that a system can be taught to recognize. They can be taught to recognize anything from animals and plants to foodstuffs and machines and can therefore be used in a wide variety of fields. A paper published in 2019 describes an experiment to try and identify which social media network or instant messaging app an image originates from [3]. This can help the police to trace images back to their origin which can be useful when dealing with cybercrimes such as cyberbullying, harassment and even cyber terrorism [3]. Not all applications for this technology have to do with crime-fighting, for example a Chinese team created a system that would recognize specific types of fruit flies [4]. These pests are responsible for a great amount of damage to China's pumpkin industry and are considered a highly dangerous pest. By identifying the fruit flies in the entry-exit quarantine will enable quarantine workers to reduce the fruit flies harmful impact [4].

II. Problem description

The main goal of this project is to make a CNN system differentiate between pictures of dolls and pictures of humans while also adding a third category of things. This category should be added to make the system less likely to leave it to chance, otherwise it would be a 50 % chance to guess right.

The goal is to determine which of the given parameters would make it harder for the CNN system to complete the classification tasks successfully.

Four types of issues were devised for the project.

1) Two different levels of CNN architecture: one classic and one rudimentary.
2) Lighting: an external light source creating defined shadows in the images.
3) Low-resolution images: three levels of low resolution.
4) Digital enhancement: brightness, contrast and negative.

Specifications

- The pictures have to be of similar make and proportions so that the system won't learn a feature unrelated to the intended motif and therefore "cheat" when testing.
- All images should be taken against a plain background.
- This third category should contain images that differ greatly from the doll category, like decorative objects, animal toy figures, flower vases and such.
- These items can have a "face" as long as they don't look like dolls.

III. Theory

Convolutional neural networks, CNN, is a deep learning algorithm. The structure is inspired by the biological workings of the brain. The CNN structure was inspired by an experiment by Hube and Wiesel on a cat's visual cortex [5]. They discovered that different sets of neuronal cells, in the visual cortex, got activated by horizontal and vertical lines. The cells are connected in a layering architecture. This layering technique is later used when creating the first type of CNN. The CNN architecture usually consists of several convolution layers with corresponding activation and pooling layers as well as other complementary layers. The three most important layers in the CNN structure are the convolution, activation and pooling layers which will be described in more detail in the sections below.

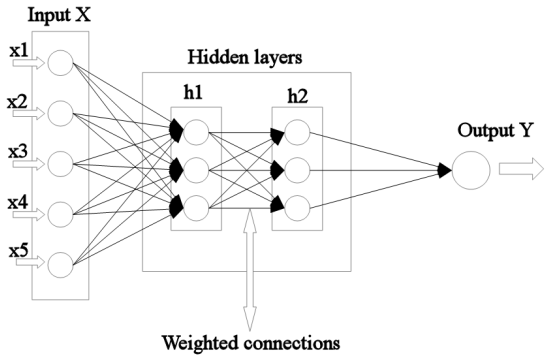


Fig. 1. A graphic representation of a multi layer neural network, with input layer, hidden layers and output layer.

Neural networks can be designed in many different ways. A single-layer neural network called a Perceptron has an input layer and an output layer and all computations in the network are made in the output layer and are therefore visible to the user [6]. A multi-layer neural network is comprised of an input layer, an output layer and in between these two layers are a number of computational layers called hidden layers. These layers are called hidden layers because their computations are hidden from the user [6], see Figure 1 for a graphical representation of the multilayer structure. The figure also shows the connections between the nodes in the different layers. These connections have a weight which is a number that shows how important that particular connection is, if the number is high the connection is important for the system and if the number is low it is less important. Just like synopsis in the brain are reinforced with use the weights become "heavier" during the training for the important connections.

A. Convolution

The convolution operation can be used for 1 to 3-dimensional inputs, where a 2D input represents an image and a 3D represents a video. The CNN system was designed for a 2-dimensional input and is therefore mostly used for image processing, taking an image or a hidden layer as its input. The hidden layer is the output of an earlier convolution

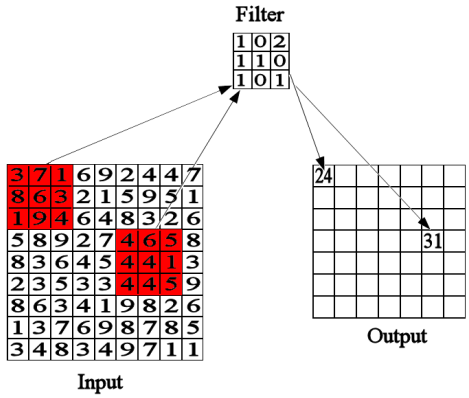


Fig. 2. A representation of the convolution operation.

in the system. The number of filters in the convolution layer will determine the depth of the new hidden layer created by the convolution operation. The system can use both grayscale images or color images the only difference being the color channel depth in the image definition layer. Where grayscale images have a depth of 1 and color images have a depth of 3. To find different features in the picture a filter, also known as a kernel, is used. The filter is defined by a filter-box that must be smaller than the input and have the same depth to operate on the whole image. The convolution operation used dot product between the input values and the values in the filter-boxes to create a hidden layer. The convolution operation will work best if the values, in the input structure, is between 0 to 1, this makes the convergence occur faster and speed up the learning. The filter-box will move through the whole image from the upper left corner down to the lower right corner see Figure 2. The stride determines the length of the step as the filter-box moves through the image. If the convolution layer has a stride of one the filter will move one pixel at the time through the image, and two pixels if the stride is 2 and so forth. Every filter is designed to find a different feature in the picture, the output from every filter is called a feature map. The number of filters will result in the depth of the new hidden layer created by the operation. The filter in the first layer registers rudiment features and the later layer registers more complex features. For more information on the convolution layer see [5].

B. Batch Normalization Layer

The batch normalization layer normalizes all elements in the input, this is done for every mini-batch. This will divide the pixel value of the image from a number between 0 and 255 (for RGB images) to a value between 0 and 1. Using a normalization layer will speed up the convolution layer and make it less sensitive [7].

C. Padding

Padding is used to avoid losing data from the edges of the picture or the input. When a convolution operation has been

performed the height and width of the output will be reduced compared to the input. This can result in data loss along the edges of the image or other input. By adding a layer of zeros along the edges of the image the filter-box can operate to the very edges of the image resulting in a complete covering of the image. This layer of zeros is called padding and can be determined manually or with predetermined models like half-padding or full-padding. The padding that will preserve most data is half-padding. Half-padding will have a layer of zeros in the size determined by equation(1) [5].

$$Z = \frac{(F-1)}{2} \tag{1}$$

Z is the number of zeros added around the edge. F is the width/height of the filter-box. This will allow half of the filter-box to stick out over the edge of the image. This will also cancel out the decrease of the output size as opposed to a convolution operation without padding. When padding is not used it is called valid padding. Another type of padding is full-padding. Full-padding will allow almost the entire filter-box to stick out over the edge. Full-padding will also increase the size of the output.

D. Pooling

The pooling operation will create a new layer with the same depth as the input layer. Pooling creates a small box with the size PxP, which typically is 2x2. The box will never have a depth bigger than one. The box will move over the picture from the upper left corner to the lower right corner. Unlike the convolution operation, the pooling layer will go through every feature map of the input and create a new feature map. This is why the output will have the same depth as the input. The stride for pooling is usually 2 with a box size of 2x2 [5]. If the stride size is 1 it will not reduce the size of the input but if the size is greater than 1 the size of the layer will be reduced. This will make the information more concentrated and the size easier to work with. Max-pooling is a pooling operation that will take the biggest value in the area that the box covers and save it for the output. Average pooling is another type of pooling but it is not commonly used. Pooling is used to help the system to learn how to find an object no matter where in the picture the object is.

E. Activation layers

1) ReLU: An activation layer is used after the convolution layer to know which neurons are active and which are dormant. The most used activation layer for deep learning is called ReLU, rectified linear unit, because it is faster and has better accuracy than the other activation layer alternatives like tanh and sigmoid.

2) Softmax: Softmax is an activation layer that is used at the end of the model's layer structure when trying to solve a categorizing problem. Softmax uses probability to determine which category the image belongs to after all computations are done.

F. Fully connected layer

Fully connected layers, also called dense layers, can be used for different applications in the CNN architecture such as segmentation and classification. It is most commonly used for classification tasks because it reduces the huge number of parameters down to the number of classification categories. This can be done in steps with more than one fully connected layer to increase the power of computations. Every feature of the final computational layer before the fully connected layer is in turn connected to each hidden state in the fully connected layer resulting in a large number of parameters [5]. If a system contains multiple fully connected layers these layers will as the name suggests be fully connected. For example if the system has two fully connected layers with 200 hidden units each the result would be 40 000 connections with their corresponding weights.

G. Loss function

The loss function determines how well a model is working. If the loss function has a high value the model is underperforming but if the value is low the model is performing well. There are many different kinds of loss functions. The loss function is chosen depending on what problem the model is trying to solve. Cross entropy loss is a common type of loss function when the model is trying to categorize its inputs.

H. Optimazer

The optimizer works together with the model's loss function to improve the model. The optimizer will use the loss function and adjust the weights of the model if the loss function is too high. Like the loss function, there is a lot of different optimizers like Adagrad, RMSprop and Adam. Adam is an accepted optimizer used to train neural networks.

I. Tensorflow

Tensorflow is an open-source library for machine learning models, written in the programming language python. It is easy for beginners in machine learning to create models since they already have prepared building blocks.

J. Common settings

A convolution neural network commonly uses square-shaped pictures, but can also take other shapes like rectangles. If an asymmetrical shape is chosen by the user then the parameters in the layers must be adjusted to accommodate the shape, like stride length, filter-boxes and padding. The most common setting for the convolution layers in CNN is to have small filter-boxes and a large amount of filters. A filter-box usually has a height/ width of 3 or 5. Small filter-boxes allow for a more detailed "look" at the image often leading to better results. The quantity of filters in the convolution layer is often a number of the power 2 because this will result in a more efficient process. It is also common to double the quantity of filters for the next convolution layer. The stride length is typically one in the convolution layer, though default settings

can vary between platforms. A larger stride length can help to reduce overfitting. This can be useful when the user has a limited amount of data power or memory. The pooling layer usually has a box of 2x2 and a stride length of 2, to reduce the size of the input.

K. Training of model

When training a CNN model many parameters can be altered and adjusted to give the model the desired functionality. Among these parameters are the epochs and the mini-batch size. One epoch equals when the model has trained on all the images in the training set 1 time, therefore multiple epochs are used to train the models. The mini-batch divides the images in the training set into batches of a predetermined size that are then given to the model in turn for training, when the model has trained on all the mini-batches it has completed one epoch.

IV. Method

The method is divided into two major sections, the Dataset and the Programming, which in turn are divided into two sub-sections. The dataset or data collecting section is divided into Photography and Digital enhancement, explaining the different steps taken when collecting and creating the datasets. The programming section is divided into Matlab and Tensorflow, describing the layout of the rudimentary Matlab model and the classical Tensorflow model.

A. Database: Photo session setup and lighting

The images used in this project were taken with a digital system camera on a tripod stand and then exported to the computer for digital enhancement and resizing. The camera takes rectangular pictures with the default image size of 5184x3456 pixels, see Table I, because the image motifs wary in size all pictures were cut down to squares and resized to the greatest pixel value commonly shared by all images.

TABLE I
CAMERA SETTINGS

Camera	Canon EOS 650D
Lens	Canon EFS 18-135 mm
Filter	DHG-UV 67 mm
Ring light	Konig KN-RL60 5500-6000 K
Automatic focus	on
ISO	800
Aperture value	F7.1
Exposure compensation	-1
Light measurement method	evaluating
Motif mode	auto
White balance	auto
Image default size	5184x3456 pixels

In order to use and compare the images from a dataset, the images must be taken in a similar if not identical way. This could mean one photo session with several participants in the same place with the same conditions. Therefore only one light source was used at a time to give the pictures a distinct property and all photo locations were made completely dark to minimize light contamination that could alter the properties of the picture. This becomes very important because of the current Covid-19 outbreak, this resulted in a greater number of photo sessions to minimize the number of people gathering at any one time, so by standardizing the photo conditions the photos could be taken in different locations but still have the same properties. The participants were asked to sit straight

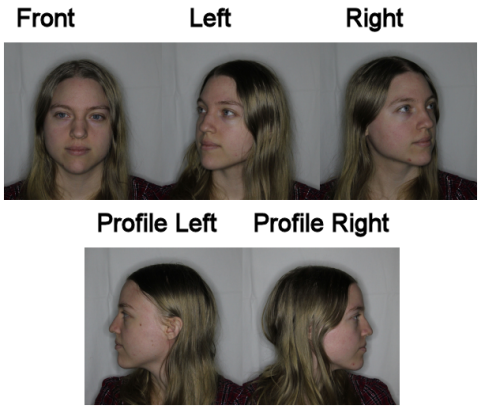


Fig. 3. The different positions of the face for dolls and humans.

and relaxed in front of the camera looking straight ahead. The camera stand's height was then adjusted so that the focal point was positioned on the person's nose and the zoom function was used to get a close-up of the face and shoulders. Then the ceiling lights were turned off and then the ring light, see Table I, was switched on. The participant was asked to look straight ahead then turn their head 45 degrees left and right (marked out with dots on the walls) and then turn their heads 90 degrees also left and right, see Figure 3.

Fig. 4. A sample of doll images used in the project.

The ring light was then switched off and the LED work light was switched on, the participant was asked to repeat the earlier face positions first with the light above them (held in place by an assistant). Then with the light on the left side, parallel to the person's face and then on the right side of the person. The work light is quite strong and the user's manual clearly states that one shouldn't look straight into the lamp, therefore the 90 degree profile picture was canceled because of the safety and comfort of our participants [8].

The doll pictures were taken in much the same way as the human pictures. Front, 45 degrees and 90 degrees both left and right, the only difference being that the 90 degree image with the work light was also taken because there are no potential health issues with the dolls, they don't feel discomfort facing the bright light. Figure 4 features a selection of dolls used.

The images for the thing category were taken in a similar way except instead of turning the objects to the specific angles only one picture was taken with the ring light and three pictures with the work light see Figure 6. This decision was made because many of the objects were symmetrical and would look the same if turned so a larger number of objects were used to compensate for this, Figure 5 shows a selection of items used.

Fig. 5. A sample of the different type of things in the thing category.

B. Database: Digital enhancement

All digital enhancement in this project was conducted in the PC program Jasc Paint Shop Pro 9. Therefore all brightness and contrast constants will be specific to this program. Paint Shop Pro 9 also has a ready made function for making a negative image which was used to make the negative images used in this project. Paint Shop Pro also has functions for resizing and cutting, which were used first for the initial squaring and resizing but also for the low-resolution datasets. The different dataset sizes can be seen in Table II.

The smallest doll used in this project was about 8 cm tall and when cut resulted in a picture of 1696x1696 pixels when cut to the specific configuration of head and shoulders. Therefore the resized size was determined as 1600x1600 pixels for all images. Because this doll was the smallest there wouldn't be a problem with any of the other cropped pictures, of the bigger dolls and humans, being smaller than 1600x1600 pixels.

An even number was preferable because the low-resolution images were created by resizing down to the desired level and then enlarging the image again to its original dimensions. This process was simplified by having a common image size that was easy to divide because otherwise, the image after the final enlargement wouldn't have the same size as the original because they weren't evenly divided with the pixel size. The low resolution levels were set to 1% (16x16 pixels), 5% (80x80 pixels) and 10% (160x160 pixels), see Figure 7.

TABLE II
NUMBER OF PICTURES AND DATASET SIZES

People	24
People pictures	120
Dolls	24
Doll pictures	124
Things	124
Thing pictures	124
Normal (N) dataset size	368
Light (L) dataset size	1008
Brightness + Normal (BN) dataset size	736
Light + Normal (LN) dataset size	1376

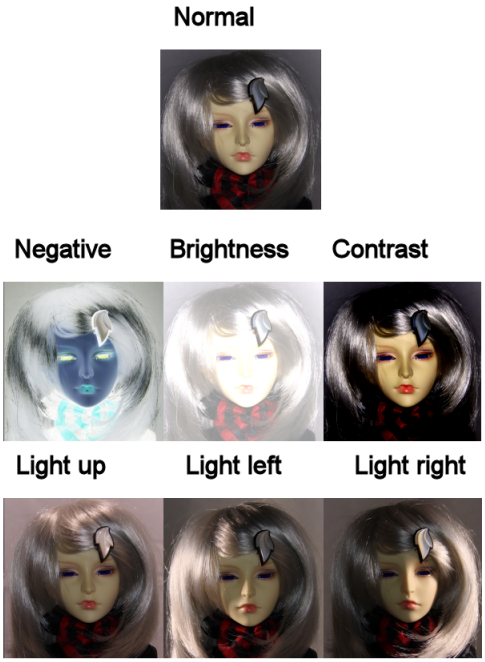


Fig. 6. A sample of all the different condition from digital enhancment.

In Paint Shop Pro 9 the brightness function can vary between -255 to 255, where -255 equals in a completely black picture and 255 equals in a completely white image. For this experiment 150 was used for the brightness in order to make the picture bright enough to dull the features of the image without removing them. The contrast values could be set between -100 and 100, where -100 equals a gray image and 100 results in a completely black and white image. The contrast value was set to 40 for this experiment as it deepened and darkened the colors without blacking then out, examples are provided in Figure 6.

Fig. 7. Low-resolution sample images.

C. Sessions

Because of the amount of sessions made in the experiment all sessions were given a unique descriptive name to tell the sessions apart. Table III shows the names for the sessions and how the names correspond to the training and testing. The first letter or series of letters, like LR1 (Low Resolution 1), in the session name is the training set, then the letter after "to" is the testing set. In the lower section of Table III the training is described as BN or brightness + normal, this means that the brightness set and the normal set were combined into a single training set.

D. System: Matlab

The model for the CNN architecture in Matlab is shown in Figure 8. The image input layer takes the height and width value of the picture and the color channels, which is 3 if it's a color image and 1 if it's a grayscale image. The images used in the Matlab program have a size of 1600x1600 pixels and color channel 3. The model has three Convolution layers, three batch normalization layers, three ReLU layers, two Max-pooling layers, a fully connected layer, a softmax layer and a classification layer. The first Convolution layer has a filter

```
layers = [
    imageInputLayer([1600 1600 3])

    convolution2dLayer(16,3,'Stride',8,'Padding','same')
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(32,3,'Stride',16,'Padding','same')
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(64,3,'Stride',32,'Padding','same')
    batchNormalizationLayer
    reluLayer

    fullyConnectedLayer(3)
    softmaxLayer
    classificationLayer];
```

Fig. 8. The CNN layers for the Matlab model.

box of 16x16 pixels and a stride length of 8 pixels, this layer uses 3 filters to learn about the image. The activation layer is of Relu type. The Max-pooling layer has a box size of 2x2 pixels and a stride length of 2 pixels. The second Convolution layer has a filter box 32x32 and a stride of 16. This is once again followed by a Relu and a Max-pooling layer with the same properties as before. The third Convolution layer has a filer box of 64x64 pixels and a stride of 32. This is followed by the fully connected layer where 3 is the number of classes in the system. Then there is the softmax layer and least the classification layer which prepares the system for classification in the classify function.

The Matlab system was trained on 80 % of the dataset and the training ran for 40 epochs to reach an acceptable mini-batch accuracy. The mini-batch accuracy for all simulations exceed 90 % but is usually between 99 and 100 %. The system then tested on the whole dataset, so there would be 20 % of the images that it had not seen before as well as the 80 % it had seen before.

TABLE III
NAME OF SIMULATION SESSIONS WITH EXPLANATION

Session name	Explanation
NtoN	Trained: Normal Tested: Normal
NtoB	Trained: Normal Tested: Brightness
NtoC	Trained: Normal Tested: Contrast
NtoNe	Trained: Normal Tested: Negative
NtoL	Trained: Normal Tested: Light
NtoLR1	Trained: Normal Tested: Low resolution 1
NtoLR5	Trained: Normal Tested: Low resolution 5
NtoLR10	Trained: Normal Tested: Low resolution 10
BtoB	Trained: Brightness Tested:Brightness
CtoC	Trained: Contrast Tested: Contrast
NetoNe	Trained: Negative Tested: Negative
LtoL	Trained: Light Tested: Light
LR1toLR1	Trained: Low resolution 1 Tested: Low resolution 1
LR5toLR5	Trained: Low resolution 5 Tested: Low resolution 5
LR10toLR10	Trained: Low resolution 10 Tested: Low resolution 10
BNtoB	Trained: Brightness + Normal Tested: Brightness
CNtoC	Trained: Contrast + Normal Tested: Contrast
NeNtoNe	Trained: Negative + Normal Tested: Negative
LNtoL	Trained: Light + Normal Tested: Light
LR1NtoLR1	Trained: Low resolution 1 + Normal Tested: Low resolution 1
LR5NtoLR5	Trained: Low resolution 5 + Normal Tested: Low resolution 5
LR10NtoLR10	Trained: Low resolution 10 + Normal Tested: Low resolution 10

E. System: Tensorflow

The CNN model for Tensorflow's architecture is shown in Figure 9. The model is programmed after Tensorflow's example on how to model a convolutional neural network [9]. Images for the input have the size of 800x800 pixels, which was chosen to make the amount of data smaller. The color channel will be three since the pictures are in colors. The first step in the model is to normalize the value of the input by dividing the values by 255 to get the input value between zero and one.The model has two Convolution layers, two Max-pooling layers and two Dense layers (Fully connected layers). The first Convolution layer used 16 filters to learn about the image and use a 3x3 filter-box with stride 1. The padding is 'same', which means that the output gets the same height and width as the input. All activation layers use ReLU. Max-pooling layer use stride 2 and have a box size of 2x2. The second Convolution layer used 32 filters to learn about the image, the second Convolution layer has the same filter size and stride as the first. Before the Dense layer, the feature maps

```
model = Sequential([
  layers.experimental.preprocessing.Rescaling(1./255, input_shape=(img_height, img_width, 3)),
  layers.Conv2D(16, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(32, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Flatten(),
  layers.Dense(64, activation='relu'),
  layers.Dense(num_classes)
])
```

Fig. 9. The CNN layers for the Tensorflow model.

need to be flattened. The output will be a vector 1280000x1. The Dense layer has 64 neurons and a ReLU activation layer. The last layer will take the vector down to 3 neurons for the output.

To train the model every dataset is divided into a training set, validation set and a test set. In the training set, only 60 % of the pictures are used, 20 % are used for the validation set. The model is tested on the whole set, with 20 % unknown pictures. The model was trained with 10 epochs with a batch size of 10.

The model has the cross-entropy loss function and uses the optimizer Adam, see Figur 10.

```
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
```

Fig. 10. The compiler for the Tensorflow model.

V. Results

Tabel IV shows the accuracy spans for the different simulations in both Matlab and Tensorflow. The spans are determined using 5 simulations of every group and looking for the lowest and highest accuracy after testing. This method was used because even if the system has the same training results the test results will vary in the group creating an accuracy span. Some positions in Tabel IV only have one value which is because in these groups the system guessed on a single category for the whole set resulting in an accuracy of 33,7%. The positions with the dashed lines represent groups that were too large for the classical model to handle without more advanced hardware.

TABLE IV
THE RESULTING ACCURACY SPANS FOR THE MODELS

Session Name	Accuracy span Matlab [%]	Accuracy span Tensorflow [%]
NtoN	95,92 - 98,10	99.18 - 99.73
NtoB	33,97 - 57,07	33.70
NtoC	86,41 - 94,29	83.70 - 95.65
NtoNe	21,47 - 35,33	32.34 - 35.05
NtoL	48,51 - 67,66	79.46 - 81.25
NtoLR1	86,68 - 95,11	95.38 - 98.10
NtoLR5	94,29 - 98,64	97.55 - 99.46
NtoLR10	94,29 - 98,91	99.18 - 99.73
BtoB	91,85 - 97,55	33.70 (44.84 Epoch 20)
CtoC	89,95 - 97,01	98.37 - 100
NetoNe	96,20 - 98,64	96.74 - 98.91
LtoL	95,93 - 98,31	—
LR1toLR1	90,76 - 98,10	98.64 - 98.91
LR5toLR5	94,84 - 98,64	99.46 - 100
LR10toLR10	96,20 - 98,10	98.64 - 99.64
BNtoB	78,53 - 95,11	74.73 - 82.88
CNtoC	96,47 - 99,18	99.46 - 99.73
NeNtoNe	92,12 - 97,01	84.78 - 97.01
LNtoL	96,63 - 97,72	—
LR1NtoLR1	97,83 - 99,46	99.18 - 99.73
LR5NtoLR5	98,64 - 100	99.73 - 100
LR10NtoLR10	96,74 - 99,73	99.46 - 99.73

When trained on the normal dataset and tested on a different dataset both systems had the best results on normal-, low-resolution 5% - and low-resolution 10%-sets. The programs had difficulty when tested on the brightness-, negative- and light sets. The graphs in Figure 11 and 12 shows the training

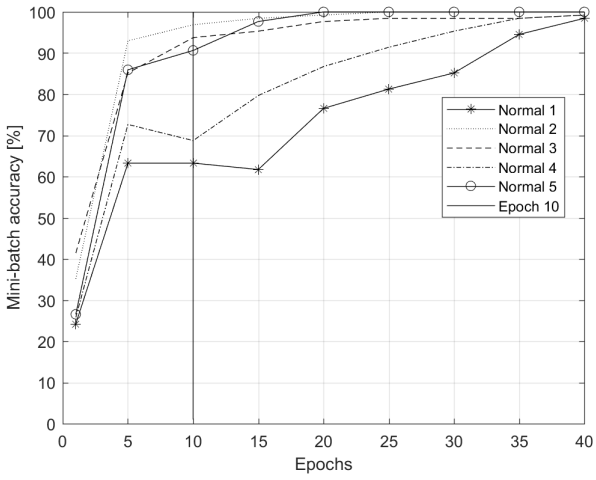


Fig. 11. Five training curves form the Matlab model.

curved on the normal dataset for the respective systems. The Matlab system learns at a slower pace than the program in Tensorflow. At 5 epoch only one of Matlabs training curves have an accuracy higher than 90 %, and all but one of the training curves from Tensorflow have exceeded 90 % accuracy. If the program gets to train on a dataset it will have a higher accuracy when tested on the same dataset. The program in

Tensorflow does not learn from training on the brightness-set, it will need more epoch to learn as can be seen in Table IV.

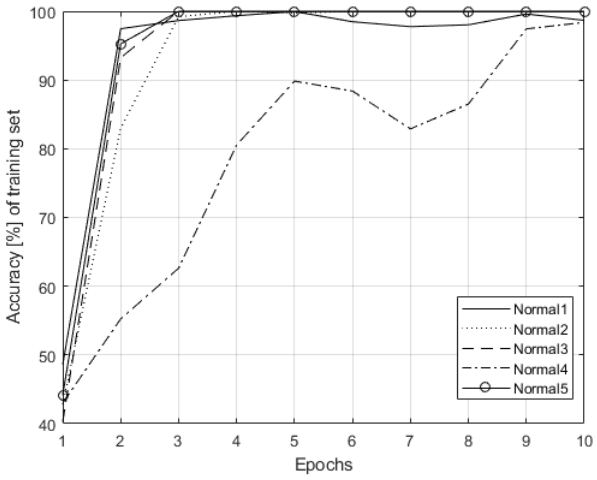


Fig. 12. Five training curves form the Tensorflow model.

The Matlab program had the lowest accuracy when trained and tested on the contrast-set, it also had a lower accuracy when trained on the brightness-set. Both programs had the highest accuracy when trained on low-resolution 5%. Overall the Tesorflow program found the brightness-set most difficult. The program in Matlab also found the brightness-set hard but not as much as the Tensorflow program. Both programs found the low-resolution 5%- set to be among the easiest to categorize.

VI. Discussion

The two systems used in this project were deliberately made to be different from each other. The system made in Tensorflow was designed in the classic way of making CNN programs, small filter boxes, short strides and many filters. This model also has several help functions like validation sets and optimizer functions.

The system made in Matlab was made to be a lot less refined, with bigger filter boxes that doubled in size for every convolution, a stride length that instead of 1 was half of the filter box size. Instead of progressively more filters as in Tensorflow, the Matlab version only had 3 filters for each convolution layer.

The Matlab version runs for 40 epochs instead of Tensorflows 10 for it to reach an acceptable training result as seen in Figure 11. The vertical line shows the 10 epoch mark and easily shows that this amount of training would result in poor performance of the system while this amount of training is more than enough for the Tensorflow system see Figure 12.

A. Brightness

Both models had problems when it came to categorizing the brightness dataset. This is probably because the contours and features of the objects are less defined. The model in Tensorflow had a hard time even identify the object when trained on the brightness set. This problem was not found in the Matlab model. This difference between the models can be because of the lesser depth of the Matlab model. The Matlab model probably uses simpler patterns to differentiate between the categories, and the Tensorfolw model has more complex patterns. The rough patterns used in the Matlab model were easier to find in the picture even though they were less defined giving an overall better result.

B. Shadows and contrast

Shadows obscure the features of an object making it harder for the models to recognize the object in the image. If the training contains this kind of shadowy image the model will learn and perform just as well as on normal images. Shadow can be created with other light sources but can also occur with higher contrast. The Matlab model had lower accuracy than expected when it trained and tested on the contrast set. This may be because the contrast in the image heightened the small shadows already was in the image. This problem was not discovered in the Tensorflow model which instead better results when tested on the contrast data set.

C. Categorization errors

It quickly became apparent when running the simulations that some images were reoccurring as errors in the system. Some of the thing images would often be returned as dolls and some doll images would be returned as things both these categories would also return human sometimes but a lot less frequent. Figure 4 shows a selection of dolls that the systems struggled with. Between the two models, the Matlab model had the most errors because it was overall not as accurate as the Tensorflow model. The interesting observation is that the two systems didn't always have the same errors. Matlab struggled most with dolls 2, 3, 4, 7 and 8 from Figure 4, while Tensorflow struggled with dolls 5, 8 and 9. Doll 5 was most of the time defined as a human while dolls 8 and 9 were always categorized as things. In Matlab doll 2 was almost always defined as human while 3, 7 and 8 were things. Doll 4 would usually switch between thing and human. We presume it is because only one eye is visible on that particular doll. Dolls 3 and 9 are both made of quite shiny plastic leaving odd light reflections on the surface. The reflections are similar to the reflections in the glass objects in the thing category which may be a reason why they are categorized as things. Doll 8 is also categorized as a thing in both the systems emphasizing that something with the figure is very hard for the systems to see or interpret correctly. The doll has a nicely defined face and the only feature that we can identify as a potential problem is her mask. We guess it's the mask but can't say without more tests of different dolls in masks.

When it comes to the thing images that were often categorized incorrectly some patterns were easier to identify than others. For example in the Matlab model images that had two or more quite defined dots in the image were often categorized as dolls like thing 12 in Figure 5. We also noticed that this categorization seemed to happen often if the motif was round or had an overall rounded shape, therefore thing 7, the tennis ball was often a doll according to the model as well as the

heart-shaped pin cushion and picture frame. Then there were other categorizations that weren't as easy to understand like things 4, 5, 6 and 11. When looking a thing 5 the place where the handles of the bag meet the body of the bag the green glaze is thicker and therefore darker and presumably the system views them as dots or "eyes" and makes the decision that it's a doll. Thing 6 (the toy horse) and 11 (the aquarium plant) on the other hand are not as easy to explain. We presume that the system sees patterns in these images that it interprets as something doll-like even if a human doesn't see it. The most confusing picture errors are things 1, 8 and 9 because these three objects are predominantly categorized as humans in both systems. Once again the systems most see something a human doesn't when defining these images into categories.

D. Low Resolution

Surprisingly both models performed very well on the low-resolution 5% dataset. This might be because the data in the image is coarser and less detailed, which makes it easier for the model to process. But there is still enough information in the image so it won't become unrecognizable. The low-resolution 1% dataset seems to be right on the line of where it starts to become too little. Even the low-resolution 1% dataset performed very well and much better than expected because this is the set that would be the most difficult for a person to identify.

VII. Conclusion

The Matlab model and the Tensorflow model are different from each other but they still have problems with the same type of datasets. They had different degrees of difficulty with the different data types but it was still the same types that caused problems. Both models found the same type of data set easy. This is not so strange considering that the two models are both CNN models.

- Round and overall rounded shapes are interpreted by the systems as dolls.
- The contour and shape of the object in the image is the most important factor for the model when categorizing the image. Heightening the brightness in the image will make it harder to see the shapes.
- The color of the object does not matter, it is the contour and shape of the object that is important.
- Dolls with more adult features and clear facial features have a higher chance to be mistaken for a human.
- A lower resolution of the image can make it easier for the model to categorize the image right.
- Shadows can make it harder to recognize the object on the image if the model has not trained on it.
- If the model has trained on a specific condition it will have a high accuracy when tested on the same condition.
- The model with the most depth does not always perform the best.

Appendix A
Table of dolls in the doll category

Appendix A contains a table of the 24 dolls used in this project. the table contains a numeral index, the "sex" of the doll, a description of the doll type and a frontal image from the normal doll dataset.

Appendix B
Table of items in the thing category

Appendix B contains a table of the 124 items used in this project. The table contains a numeral index, a short description of the object as well as an image from the normal thing dataset.

Appendix C
Table of humans in the human category

Appendix C contains a table of the 24 humans who participated in this project. The table contains a numeral index, the sex of the human as well as their age. The participants did not consent to have their images published in this work so there are no reference images provided in the appendix.

Acknowledgment

The authors would like to thank Ida for letting them using her home, their supervisor Saikat for his guidance and support and Alireza and Anubhab for answering their questions. They would also like to thank all the people that participated in this project.

References

[1] (2021, Apr.) Vad är ansiktsigenkänning på facebook och hur fungerar det? Facebook, USA. [Online]. Available: https://www.facebook.com/help/122175507864081

[2] (2020, Mar.) Facial recognition. INTERPOL. [Online]. Available: https://www.interpol.int/How-we-work/Forensics/Facial-Recognition

[3] I. Amerini, C.-T. Li, and R. Caldelli, "Social network identification through image classification with cnn," *IEEE Access*, vol. 7, pp. 35 264–35 273, March 2019.

[4] Y. Peng, M. Liao, Y. Song, Z. Liu, H. He, H. Deng, and Y. Wang, "Fb-cnn: Feature fusion-based bilinear cnn for classification of fruit fly image," *IEEE Access*, vol. 8, pp. 3987–3995, December 2020.

[5] *C. C. Aggarwal, Neural Networks and Deep Learning: A Textbook*. Cham, USA: Springer International Publishing AG, 2018, ch. 8, pp. 315-328.

[6] C. C. Aggarwal, *Neural Networks and Deep Learning: A Textbook*. Cham, USA: Springer International Publishing AG, 2018, ch. 1, pp. 17-18.

[7] (2021, May) Batch normalization layer. The Mathworks Inc, USA. [Online]. Available: https://se.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.batchnormalizationlayer.html?s_tid=doc_ta

[8] (2019, May) Arbetslampa 008733. Jula AB, Sweden. [Online]. Available: https://www.jula.se/globalassets/catalog/productdocuments/manual/008733_se-no-pl-en.pdf

[9] (2021, Mar.) Convolutional neural network (cnn). Google, USA. [Online]. Available: https://www.tensorflow.org/tutorials/images/cnn

Evaluating Methods for Show-through or Bleed-through Cancellation

Andreas Åström and Erik Martin Welin

***Abstract*—The theme of this paper is show-through and bleed-through cancellation of degraded images or manuscripts. Show-through and bleed-through is when the reverse side of a manuscripts shows/bleeds through to the front side. This can severely impact readability. Four different cancellation algorithms was chosen for evaluation. The methods considered are Thresholding, ICA and two other methods, herewithin referred to as 'Dubojs' and 'ISJ4'. ICA is a blind source separation method. Dubojs is a more sophisticated thresholding method, while ISJ4 uses statistical methods for estimating parameters of a show-through model. In this paper, the different algorithms are evaluated on different types of common text degradations. The authors found that the most useful methods were ISJ4, Dubojs and ICA, with ISJ4 being the most flexible. Thresholding was found to be too simple. Dubojs needs improved segmentation to work properly, but was useful in a particular case of strong bleed-through. ICA was easy to use, but required linear mixtures to work properly.**

***Sammanfattning*—Temat för detta arbete är borttagning av 'show-through' och 'bleed-through' som påverkat läsbarheten hos skannade, eller handskrivna dokument negativt. Dessa fenomen uppstår på dubbelsidiga dokument och leder till att delar av baksidan dyker upp på framsidan och vice versa. Fyra olika algoritmer som tar bort de oönskade bidragen har utvärderats. Dessa är Thresholding, ICA och två andra metoder som kommit att kallas 'Dubojs' och 'ISJ4'. Dubojs är en mer sofistikerad Thresholding-algoritm medan ISJ4 använder statistiksa metoder för att skatta paremterar i en show-through-modell. Algoritmerna testades och utvärderades på dokument som påverkats av show/bleed-through i olika utsträckning. Författarna fann att de mest användbara metoderna var ISJ4, Dubojs och ICA, där ISJ4 var mest flexibel. Thresholding var för simpel för att vara riktigt användbar. Dubojs skulle behöva förbättrad klassificering, men var användbar i ett speciellt fall av bleed-through. ICA var lätt att använda, men krävde linjära blandningar för att fungera på ett önskvärt sätt.**

***Index Terms*—show-through, bleed-through, cancellation, ICA, thresholding**

Supervisors: Mats Bengtsson

TRITA number: TRITA-EECS-EX-2021:189

I. Introduction

There is an abundance of valuable information currently unavailable to us. Some of it is stuck in physical form throughout libraries all over the world. This information can be considered dead information. This is because it's hard to access the information and this disincentivizes people from using it.

There exists an effort to capture and digitalise this information. For example by scanning documents, or by using character recognition to convert handwriting into digital text. To succeed, a few problems have to be dealt with. One of the problems happens when the text from the reverse side of the page shows through. This can happen in the scanning process, since some of the light, from the scanner, passes through the paper [1]. This is called the show-through problem. Another frequent issue is the problem of bleed-through. Bleed-through occurs when ink from one side of a paper sips through to the other side. If the paper has writing on both sides, the ink bleed-through will degrade readability of the affected side. If show-through and bleed-through can be removed, documents can more easily be read as is, or converted to text by machines.

Both show-through and bleed-through give rise to nonlinear mixtures of images. The fact that show-through is nonlinear has, for example, been shown by [2]. Restoring the original images is a problem that can be classified as 'Blind Source Separation' (BSS). The term 'Blind' refers to the fact that the original, unmixed, images are unknown, i.e. there is little information available about the properties of the original source signals. In general, non-linear BSS is a very difficult problem and there is no clear cut method to separate an arbitrary, nonlinear mix. In practice, certain assumptions about the mixtures have to be made for separation to be possible. Statistical independence is one such assumption. It has been shown that show-through is close to linear [1]. Because of this, methods that assume linearity such as Independent Component Analysis [3] or transforming the problem to locally linear mixtures [4] have been tried with some success. The purpose of this project is to implement, compare, and evaluate, a few different methods to perform BSS on image mixtures. The methods will be evaluated with regards to how well they improve readability of the documents. Readability is difficult to quantify, so subjective criteria will be applied. Using criteria such as minimizing mean squared error has been shown to be less successful than visually assessing readability. In order to improve readability the algorithms should remove show-through and bleed-through. If the algorithms introduce distortions, the distorsions will be considered problematic if they make reading more difficult, or distract the reader, and insignificant otherwise.

The methods chosen for consideration are Independent Component Analysis, Thresholding, 'Dubojs' [5] and 'ISJ4' [2]. The method's were chosen partly based on how complex they were to implement and execute, with Thresholding being by far the simplest and the one referred to as 'ISJ4' being the most time consuming to implement and run. Another idea was that the algorithms should vary in terms of how they are implemented, as well as assumptions used. Several methods that assume linear mixtures have been tried, as well as several mixtures that rely on linear filtering, just to give a

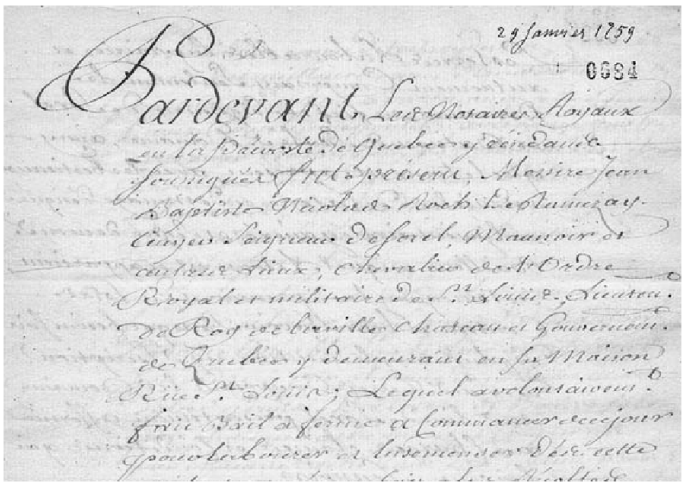


Fig. 1: Segment of bleed-through document. Source: [7]

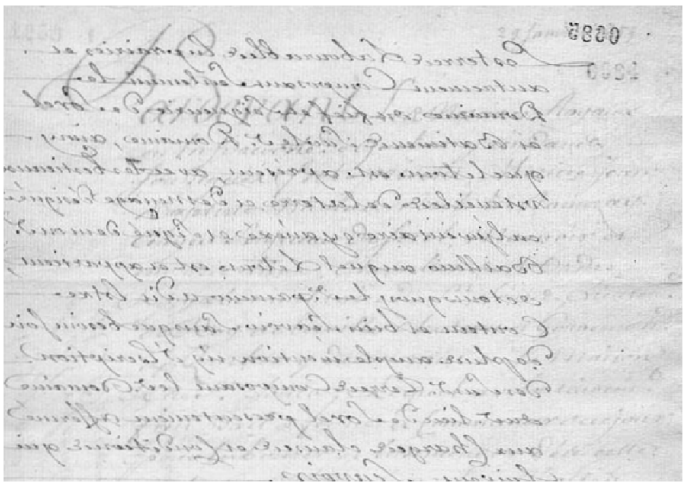


Fig. 2: Segment of bleed-through document, heavy bleed-through in the top half. Source: [7]

few examples. In the chosen batch, there is only one method that assumes perfect linearity and only one method that makes use of linear filtering. It was deemed pointless to compare methods that are too closely related.

Each algorithm will be given its own subsection, with a description, necessary theory, short discussion and results of implementation. The comparison between the algorithms will be located at the end of the report.

A. Images considered

Throughout this paper the algorithms will be compared to each other. They'll be considered based on performance under different scenarios. The scenarios considered will be:

1) Show-through
2) Bleed-through
3) Heavy bleed-through
4) irregular bleed-through
5) Heavy irregular bleed-through

The heavy prefix signifies that the show/bleed-through is as intense as that of the foreground, while the irregular prefix signifies spotty or uneven bleed-through. The images below are gathered from two databases, one from Dublin Institute for Advanced Studies [6], and the rest from Dubois [7]. Fig. 1 and Fig. 2 are examples of bleed-through. Fig. 2 also shows hints of heavy bleed-through.

Fig. 3 and Fig. 4 demonstrate show-through.

Fig. 5 demonstrates irregular bleed-through. There are also some instances inside the text where there exist irregular, heavy bleed-through. The reverse side of Fig. 5 was also used but is not shown here.

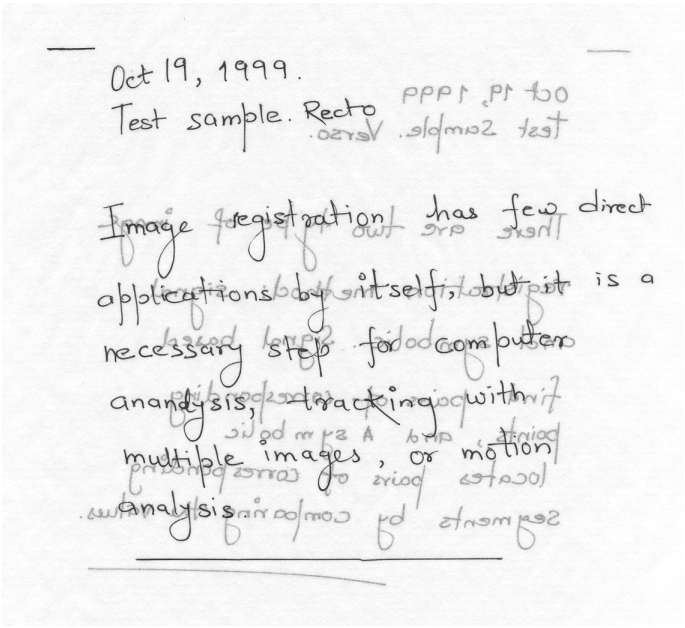


Fig. 3: Show-through image. Source: [7]

II. Thresholding

A. Theory

The most primitive form of thresholding [8] is to replace each pixel in an image with either black or white, depending on if the pixel value is greater or less than some thresholding value, t. This thresholding value t can be fixed globally throughout the image, or locally. The simplest of thresholding is as follows. If $i(m,n) < t$, let pixel $i(m,n)$ = 0 (black), else let pixel $i(m,n)$ = 255 (white). Here $i(m,n)$ is the pixel intensity of row m and the column n. This process is known as binarizing. A satisfactory value of t can be estimated a number of different ways, one of which is through Otsu's method. Other ways include manually looking at the histogram and choosing a suitable value that divides the histogram into classes.

1) Otsu Thresholding: Otsu's thresholding [9] is an adaptive thresholding method. The method is useful for bimodal images. Theese types of images are characterised by having two distinct peaks in its histogram. Otsu's method calculates a global image threshold value t by minimizing the within-class variance of the image. These classes are defined by splitting the image histogram into 2 classes. There will then be 2 clusters. The first cluster has pixel values between 0 and t, while the second cluster will have pixel values between $t+1$

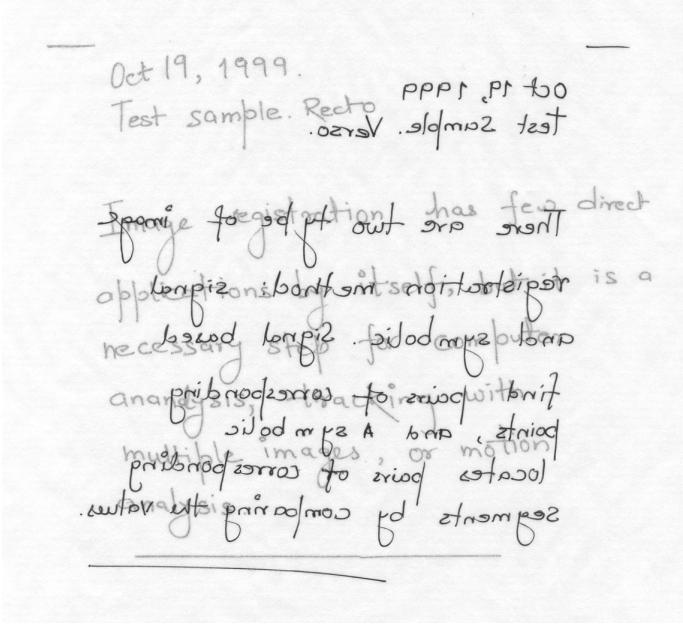


Fig. 4: Show-through image. Source: [7]

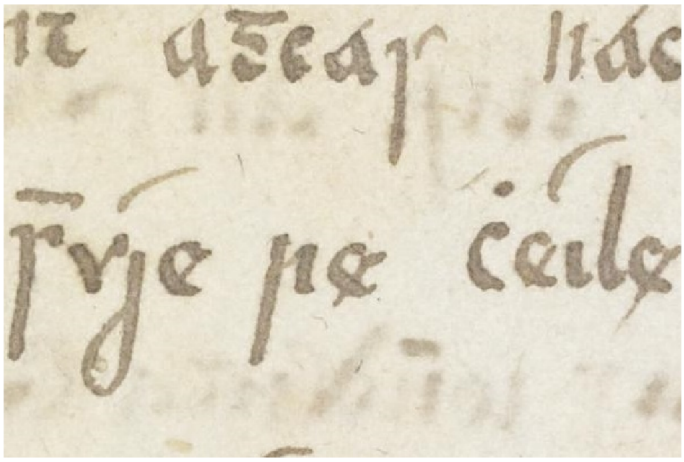


Fig. 5: Irregular and occasionally heavy bleed-through. Source: [6]

and maximum intensity value I. The within-class variance, which is to be minimized is defined below:

$$\sigma_w^2(t) = w_1(t)\sigma_1^2(t) + w_2(t)\sigma_2^2(t). \tag{1}$$

Low within-class variance is desirable because the lower $\sigma_w^2(t)$ is, the less dispersed the data in each cluster is. In equation 1, $w_1(t)$ and $w_2(t)$ is defined as:

$$\begin{aligned} w_1(t) &= \sum_{i=1}^{t} P(i) \\ w_2(t) &= \sum_{i=t+1}^{I} P(i) \end{aligned} \tag{2}$$

and can be thought of as the cluster probability functions for each respective cluster. i is defined as a certain intensity level. There are n_i number of intensity level i pixels while the whole image consists of n pixels. The probability density functions with respect to the pixel values i is typically estimated using the relative frequency

$$P(i) = \frac{n_i}{n}.$$

This could be interpreted as a histogram. To calculate the variance within the classes is done as follows

$$\begin{aligned} \sigma_1^2(t) &= \sum_{i=1}^{t} [i - \mu_1(t)]^2 \frac{P(i)}{w_1(t)} \\ \sigma_2^2(t) &= \sum_{i=t+1}^{I} [i - \mu_2(t)]^2 \frac{P(i)}{w_2(t)} \end{aligned} \tag{3}$$

where the $\mu_i(t), i = 1, 2$ is the mean of cluster 1 or 2.

B. Implementation

The image histogram were calculated with m bins. Bins refer to how many intervals the pixel intensities of the image is split up into. For most images, $m = 256$ was good. Next step was to iterate through all the possible thresholding values t, where t was restricted to be a part of the set [0, 255], i.e. the possible pixel intensities. For every t value the following was done:
The values of w_i, μ_i were computed. The inbetween-variance σ_w^2 was then determined from the newly computed w_i, μ_i values. The final threshholding value is that t-value which causes the minimization of the inbetween-variance σ_w^2. Last step was to binarize the image with the use of the final thresholding value.

C. Results

Fig. 6 was done with a histogram, with a bin size of $m = 256$. It resulted in a threshold value of $t = 188$.

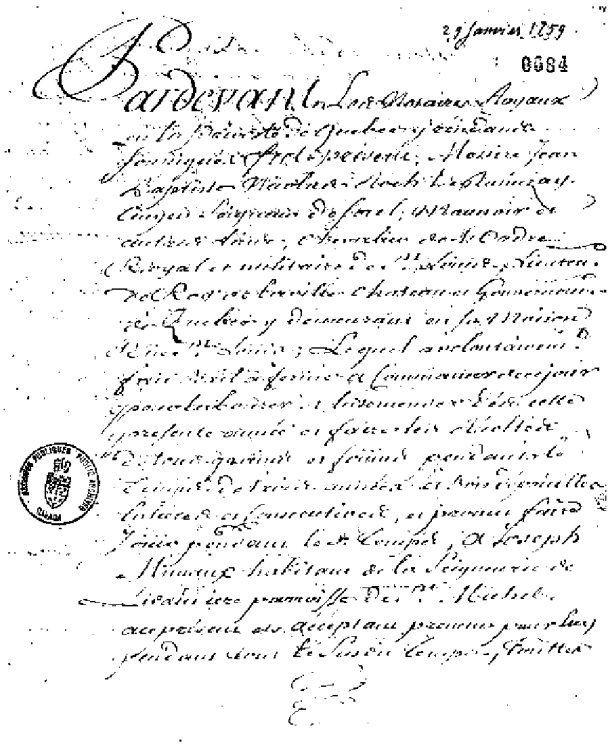


Fig. 6: Otsu's method on bleed-through document, $m = 256$

Fig. 7 and Fig. 8 were done with a bin size of $m = 3$ and $m = 256$. Thresholding values of $t = 128$ and $t = 186$ respectively were derived.

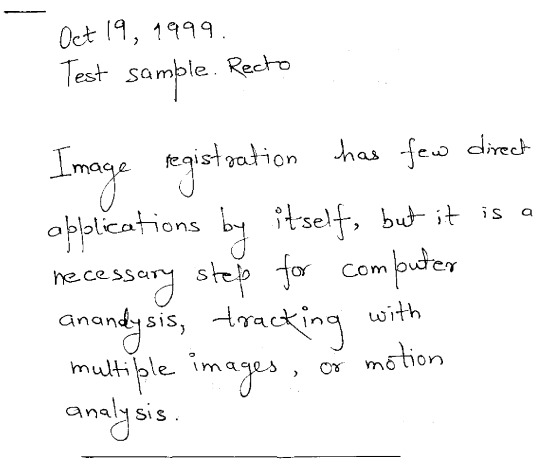


Fig. 7: Otsu's method on document, $m = 3$

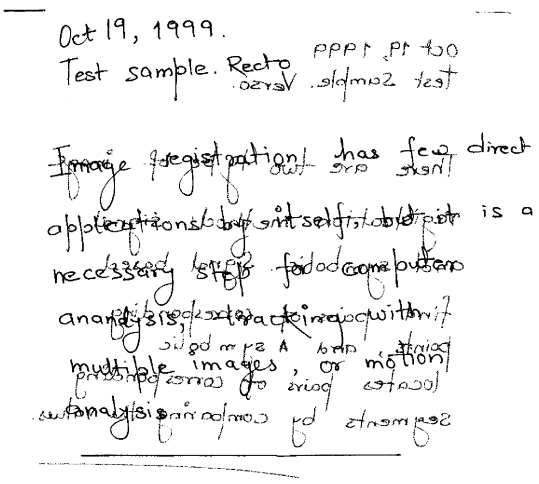


Fig. 8: Otsu's method on document, $m = 256$

D. Discussion

Thresholding methods result in an image with two clusters, pixels with values 0 and 255. It results in a binary image. This method uses only information from one page, therefore it can be expected to perform worse than other more sophisticated methods. A good way of choosing the thresholding value was through otsu's method. Otsu's method presents an automatic way of choosing a good thresholding value, in the sense that there is no human input required for tuning certain parameters. It only seeks to minimize the inbetween-variance. This makes the method very easy to use. As can be seen, selecting the histogram bin number can be the difference between a good and a bad result. This can be seen in Fig. 7 and Fig. 8, where the later figure did not succesfully remove all the noise because of its large bin size. For most images the standard value of 256 bins was sufficient. In this case however, Fig. 7 worked because with a bin size of $m = 3$, otsu's method could not be very precise. Therefore otsu's method got it right in this case, but only because of using the wrong bin size. Otsu's method did not work as intended.

Thresholding works best when the source image foreground and show/bleed-through differs in their intensity levels. Otsu's method works better when there exist a good contrast ratio between bleed-through and foreground. Since the method only discriminates based on pixel-values, if the foreground and bleed-through share the same pixel-values it will not be segmented correctly, as can be seen in Fig. 6 where there is a lot of noise left. The original image had bleed-through that in some cases was as intense as that of the foreground.

The method works well on other types of images. Notably, when the show/bleed-through is in a pixel-value cluster and the foreground is in another distinct cluster. Then they can easily be separated, as shown in Fig. 7.

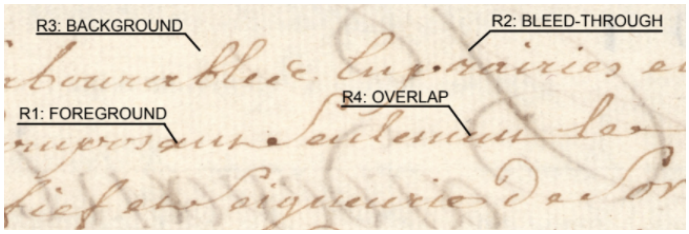


Fig. 9: The four segments considered in this algorithm. Source: Adapted from [5, Fig. 2]

III. Dubojs

A. Theory

The Dubojs algorithm is a segmentation method based upon [5] comparing the recto and verso sides of a page. Recto is latin for front and verso is latin for back. It considers these four segments,

R1: Foreground only
R2: Bleed through only
R3: Background
R4: Foreground and bleed-through overlap

See Fig. 9. The algorithm is used to classify bright areas of the pages as background. If the recto pixel is considerably more dark than its respective verso pixel, the recto pixel is considered foreground. In order to segment possible bleed-through (R2 and R4), correlation was used. In [5] an assumption was made that for R2 only (bleed-through) regions, the recto and verso image regions should be highly correlated while for overlap they should not be.

Another assumption made by [5] was that the bleed-through process may make the ink occupy a larger area on the reverse side after it has penetrated through the paper. Therefore a pixel-by-pixel comparison may not necessarily be successful and comparing areas using local windows might be a better approach. Local windows are defined to be

$$W_P = \left\{ (k,l) | -\frac{P-1}{2} \leq (k,l) \leq \frac{P-1}{2} \right\} \qquad (4)$$

In equation (4), P is the window size, assumed to be odd to allow for a centerpoint. $0, 0$ is the center of the window.

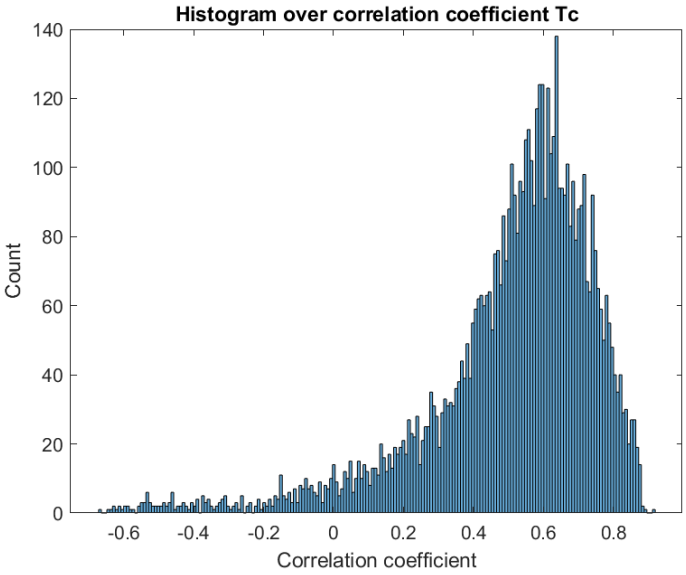


Fig. 10: Histogram of $C[m,n]$ values. Example of when $T_c < 0$ was used

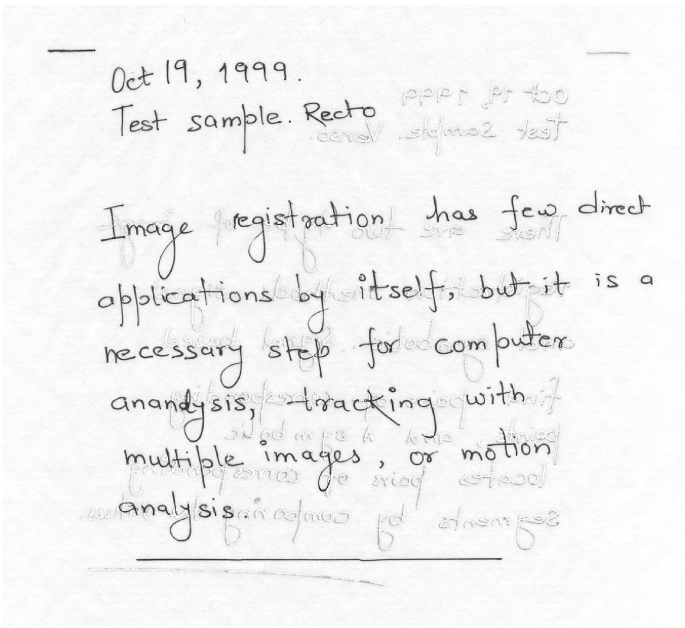


Fig. 11: Show-through removed with Dubois inpainting algorithm. Threshold value $T_c = 0$

B. Implementation

Pixels were considered background if they were at least 90% as bright as the brightest pixel in the page. Foreground was determined through the use of a local window with window size P. The darkest point in the window was considered to be representative of the centerpoint. The recto and verso centerpoint was then compared. If $f_r \leq \alpha f_v$ then the recto centerpoint is declared to be foreground, where α is a margin added for better results. Window size $P = 5$ and $\alpha = 1.3$ were used. An alternative method is comparing f_r and f_v pixel-by-pixel with no local windows, and weigh it with α as above.
To differentiate between potential overlap (R2 and R4), a cross correlation window was defined. $C[m,n]$ is the normalised cross correlation of f_r and f_v centered at (m,n) over a QxQ square. If $c[m,n] \geq T_c$, it was assigned to region R2. The size $Q = 15$ was used, while the threshold T_c was adjusted for different images.

A histogram of the correlation coeffecients between the images, such as Fig. 10, can be used to determine a good correlation coefficient T_c value. If T_c is chosen too high, parts of the region R2 will be filtered away, meaning there will be remnants of unresolved bleed-through in the final result. A coeffecient chosen too low results in overlapping regions getting inpainted, which is not the desirable. Since the region R2 will be much bigger than region R4, and region R2 will be mostly made up of high correlation areas, it reasons that the left tail in Fig. 10 is mostly made up of region R4. Therefore a good T_c is one that lies on the tail. When R2 had been identified, [5] proposes a primitive inpainting method. This consists of making a global or local background color estimate, and replacing all the R2-pixels with this background color. As an alternative to the inpainting method suggested by [5], the region, R2, to be replaced was extended, by a certain pixelwidth while also making sure it does not encroach upon the Foreground R1 by using the classification. It then uses a background estimate for replacing the pixels in the extended area around R2 the pixel. A background pixel, in close proximity to the R2 pixel, was used as a local estimate of the background intensity for some images. For others, an estimate of the statistical distribution of the background pixels was found using the histogram and then regions were inpaiting using pseudo-random samples from the estimated distribution. Pixel intensities for all images considered were approximately normally distributed, which made this method simple to use.

C. Results

All figures presented herewithin were done with pixel-by-pixel comparisons, with the exception of Fig. 13, where local windows was used. Fig. 11 shows the result of the original inpainting method. Remnants of the previous show-through can be seen. Fig. 12 uses the proposed inpaininting algorithm instead.

Fig. 13 shows the results when local windows was used.

Fig. 14 shows the algorithm used on heavy bleed-through.

Fig. 15 shows the proposed inpainting algorithm that can be utilised when the background is irregular and spotty. This time the background intensity estimate consisted of 35 pseudo-random samples from a normal distribution with mean of 0.87 and sigma of 0.02. The pixel intensities had been rescaled from the interval [0, 255] to [0,1].

D. Discussion

That assumption underpinning the need for local windows method did not seem necessary. In fact, results were noticeably worse with local windows being implemented, as shown in Fig. 13.
An assumption was made that the bleed-through recto and

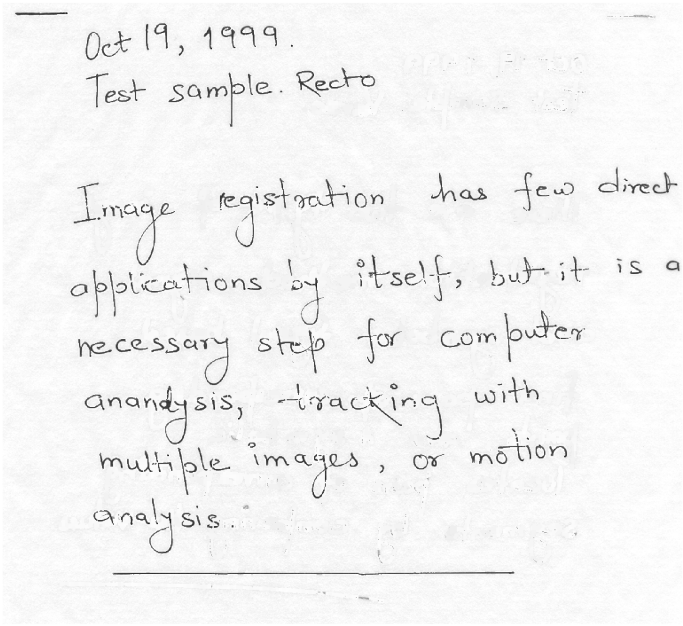


Fig. 12: Show-through removed with proposed inpainting algorithm. Threshold value $T_c = 0$

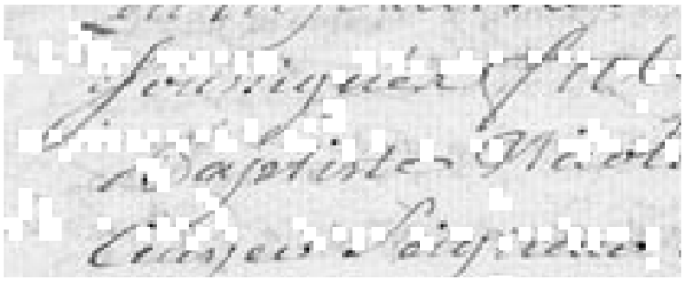


Fig. 13: Algorithm with local windows. Cropped image of bleed-through document after the algorithm had been applied

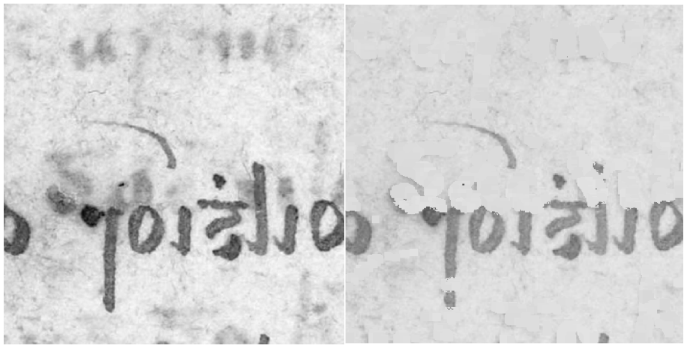


Fig. 14: Montage of bleed-through removed with the algorithm

verso image regions should be highly correlated, while for overlap they should not be. As can be seen in Fig. 10, there exist no clear divide between bleed-through and overlap regions. Should one put a cutoff value at some point, there will be information of both R2 and R4 that will go missing. Therefore using correlation to separate R2 and R4 regions was not a clean solution. Although when the histogram had long tails, such as in Fig. 10, T_c could be chosen such that it mostly

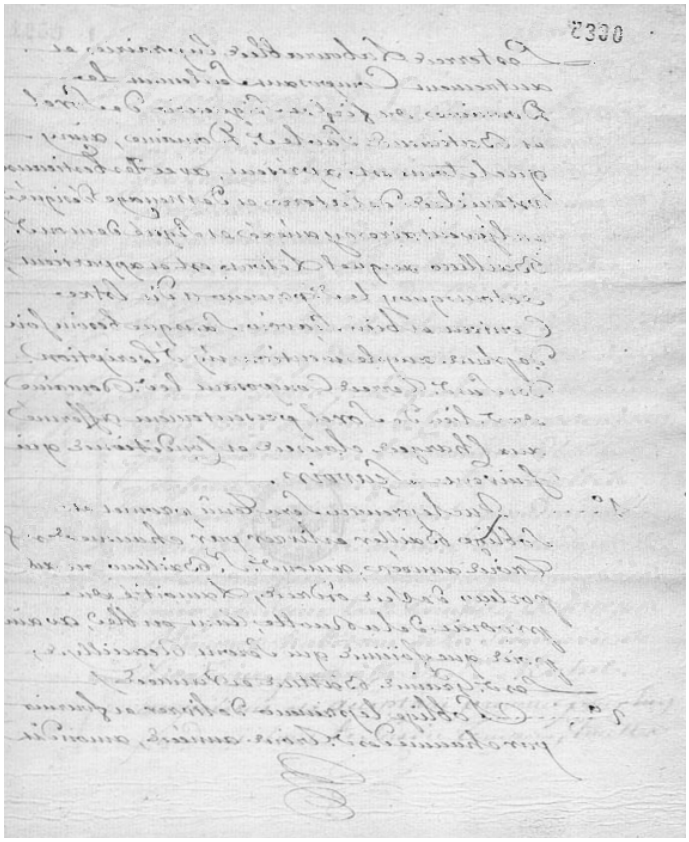


Fig. 15: A lot of bleed-through removed with proposed inpainting algorithm. Threshold value $T_c = -0.4$

filtered the R4 regions, but yet there would also be a little bit of the R2 regions being filtered as well.

When the inpainting method that painted each pixel was used, a contour around the removed sections could be spotted, see Fig. 11. Because of this, the unsolicited, inpainted, R2-areas of the document was still readable, even after the algorithm had been run.

In order to solve this problem a method that inpainted an area around the R2-pixels was tried. This method didn't work because it removed foreground pixels in places where R2 and R1 were close to each other. The removal of foreground problem, in turn led to the development of the inpainting method described in previous sections where the classification of foreground was used in the inpainting process.

When all background pixels of an image have similar intensity, or one particular intensity is very prominent, it works well to use one single intensity in the inpainting, i.e. set all unwanted pixels to the same intensity, be it a global, or local, estimate of background. As was used in Fig. 12. However, in old documents the background intensity is typically more variable, because of aging or discoloration. In this case, setting the inpainted pixels to a single intensity makes them stand out too much against the variable background intensity, which can be seen in Fig. 16.

Inpainting using intensities randomly sampled from an approximation of the distribution of background intensities, solved this problem, which can be seen in Fig. 15. In the case of Fig. 14, the inpaitning algorithm did not do a good job of

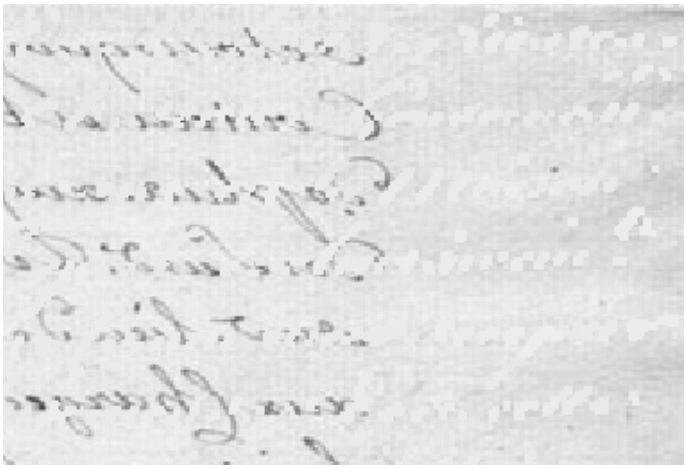


Fig. 16: hints of inpainted text can be spotted in the right column

concealing the inpainted pixels. The lettering was too thick for this to work out properly. Binarizing the image after applying Dubojs solves this problem.

The original inpainting algorithm and local windows did not adequately remove show/bleed-through. The updated inpainting algorithm, in combination with image dependent background estimates yielded results which greatly improved readability by removing unwanted elements, without disturbing the foreground.

IV. The ISJ4 Algorithm

A. ISJ4 Introduction

This algorithm is based upon physics described by [1] and mathematical work proposed in [10] and [11]. The algorithm itself is described in [2]. The algorithm wasn't given a name by its initial creator. For the purpose of this document it needs to have a name, "ISJ4" was chosen for completely arbitrary reasons. A detailed description of how the algorithm was implemented will be presented in this subsection. The purpose of this is to make the work more accessible to the target audience, i.e. undergrad engineering students.

B. Kernel Density Estimation

Kernel density estimation is a way to estimate the probability density function (pdf) of an unknown distribution given samples $x_1, x_2 \dots x_n$ from the distribution. One way (other exists) to perform Kernel Density Estimation is;

$$\hat{f}_h = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x - x_i}{h}) \tag{5}$$

where $h > 0$ is a bandwidth parameter, $K(x, x_i)$ is the Kernel function and $\hat{f}_h$ is the estimated pdf of the stochastic variable X. The Kernel function can be any function and the bandwidth parameter controls the smoothness of the estimated pdf.

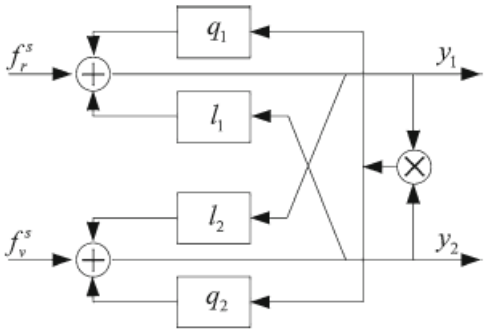


Fig. 17: Separation structure, figure taken from [2, Fig. 10]

C. Model of Show-through

As a model for show-through,

$$\begin{aligned} f_r^s &= a_1 f_r^i(m,n) + b_1 f_v^i(m,n) \times g_1(f_r^i(m,n)) \\ f_v^s &= a_2 f_v^i(m,n) + b_2 f_r^i(m,n) \times g_2(f_v^i(m,n)), \end{aligned} \tag{6}$$

has been proposed by [2]. They show experimentally that $g_i(u)$, i = 1,2 is approximately exponential, i.e. $g_i(u) = \gamma_i e^{\beta i u}$. They also show that β_i is small. Therefore, it follows that $g_i(u)$ can be replaced by it's first order Taylor Expansion. By performing the Taylor-expansion, neglecting higher order terms, and changing variables

$$\begin{aligned} f_r^s(m,n) &= f_r^i(m,n) - l_1 f_v^i(m,n) - q_1 f_r^i(m,n) f_v^i(m,n) \\ f_v^s(m,n) &= f_v^i(m,n) - l_2 f_r^i(m,n) - q_2 f_r^i(m,n) f_v^i(m,n), \end{aligned} \tag{7}$$

can be obtained from equation(6). Equation(7) is a linear-quadratic mixture, [10] proposed a recurrent separating structure (Fig. 17) that can be used to separate such mixtures. The structure requires computing of

$$\begin{aligned} y_1^{(t+1)} &= f_r^s + l_1 y_2^{(t)} + q_1 y_1^{(t)} y_2^{(t)} \\ y_2^{(t+1)} &= f_v^s + l_2 y_1^{(t)} + q_2 y_2^{(t)} y_1^{(t)}. \end{aligned} \tag{8}$$

In equation(8) the ideal images are replaced by the outputs of the structure. The structure uses the scanned images and the output of the previous iteration, y_i, to compute the next output, y_{i+1}. [2] showed that equation(7) is insufficient because it doesn't take into account the fact that the side of the paper that is facing down in the scanning process, will appear low-pass filtered on the front, because of the way light, from the scanner, is scattered through the paper. This light scattering effect was originally explained by [1]. In order to take this light scattering into account equation(7) is modified in the following manner,

$$\begin{aligned} f_r^s(m,n) &= f_r^i(m,n) - l_1 f_v^i(m,n) * \overline{H_1} \\ &\quad - q_1 f_r^i(m,n) \times (f_v^i(m,n) * \overline{H_1}) \\ f_v^s(m,n) &= f_v^i(m,n) - l_2 f_r^i(m,n) * \overline{H_2} \\ &\quad - q_2 f_v^i(m,n) \times (f_r^i(m,n) * \overline{H_2}), \end{aligned} \tag{9}$$

where * denotes the convolution operator. The separating structure in equation(8) is updated in accordance to

$$\begin{aligned} y_1^{(t+1)} &= f_r^s + l_1 y_2^{(t)} * \overline{H_1} + q_1 y_1^{(t)} \times (y_2^{(t)} * \overline{H_1}) \\ y_2^{(t+1)} &= f_v^s + l_2 y_1^{(t)} * \overline{H_2} + q_2 y_2^{(t)} \times (y_1^{(t)} * \overline{H_2}). \end{aligned} \tag{10}$$

The filters $\overline{H_1}$ and $\overline{H_2}$ model the light-scattering phenomena. If the model parameters $[l_1, l_2, q_1, q_2]$, along with the filters are known, the ideal images can be obtained from equation(9).

D. Implementation

In accordance with [2], a maximum Likelihood approach to estimating the model parameters, l_1, l_2, q_1, and q_2 was taken. Let $\overline{p} = [l_1, l_2, q_1, q_2]$, and L be the likelihood function of $\overline{p}$ with respect to the samples of the scanned images f_r^s and f_v^s. The maximum Likelihood estimate was obtained using a gradient ascent algorithm i.e.

$$\overline{p_{i+1}} = \overline{p_i} + \mu_{LR} \times \frac{dL}{d\overline{p}}, \tag{11}$$

where μ_{LR} is a called the learning rate parameter. If the learning rate is chosen sufficiently small, equation(11) should eventually converge towards a local maximum which represents the Maximum Likeliehood estimate of the model parameters. The term $\frac{dL}{d\overline{p}}$ in equation(11) can be calculated in the following manner

$$\frac{dL}{d\overline{p}} = -E[\frac{A}{J}, \frac{B}{J}, \frac{C}{J}, \frac{D}{J}], \tag{12}$$

where E is the spatial averaging operator and

$$\begin{aligned}
A &= \psi_1(s_1)(1 - q_2 s_1)s_2 + \psi_2(s_2)(l_2 + q_2 s_2)s_2 \\
&- (l_2 + q_2 s_2) - (q_2 + l_2 q_1)(1 - q_2 s_1)s_2/J \\
&- (q_1 + l_1 q_2)(l_2 + q_2 s_2)s_2/J, \\
B &= \psi_1(s_1)(l_1 + q_1 s_1)s_1 + \psi_2(s_2)(1 - q_1 s_2)s1 \\
&- (l_1 + q_1 s_1) - (q_1 + l_1 q_2)(1 q_1 s_2)s_1/J \\
&- (q_2 + l_2 q_1)(l_1 + q_1 s_1)s_1/J, \\
C &= \psi_1(s_1)(1 - q_2 s_1)s_2 s_1 + \psi_2(s_2)(l_2 + q_2 s_2)s_2 s_1 \\
&- (l_2 s_1 + s_2) - (q_2 + l_2 q_1)(1 - q_2 s_1)s_1 s_2/J \\
&- (q_1 + l_1 q_2)(l_2 + q_2 s_2)s_1 s_2/J, \\
D &= \psi_1(s_1)(l_1 + q_1 s_1)s_1 s_2 + \psi_2(s_2)(1 - q_1 s_2)s_1 s_2 \\
&- (s_1 + l_1 s_2) - (q_1 + l_1 q_2)(1 - q_1 s_2)s_1 s_2/J \\
&- (q_2 + l_2 q_1)(l_1 + q_1 s_1)s_1 s_2/J, \\
J &= 1 - l_1 l_2 - (q_2 + l_2 q_1)s_1 - (q_1 + l_1 q_2)s_2,
\end{aligned} \tag{13}$$

where $\psi_i(s_i)$, i=1,2 is the Score-functions with respect to the ideal source images s_i. The score-function is defined as the gradient of the log-Likelihood function, with respect to the parameter vector. For computational details of the results in equation(13), see [10] and [11]. The ideal images were, of course, unknown, hence the latest estimates of the ideal images y_i was used instead. The terms A, B, C, D and J in equation(13) are matrices with the same dimensions as the images used. Spatial Averaging was implemented using componentwise arithmetic mean and then convoluting the resulting 1x4 vector with a 2D-Gaussian Kernel with $\sigma_x = \sigma_y = 0.5$, where σ_m represents the standard deviation. The Kernel used was $K_{Gauss}(x, y) = e^{-((x^2+y^2)/2)}$. The convolution was used because it made the separating structure less prone to divergence.

The equation(13) requires the computation of score functions. The estimate

$$\frac{\nabla_x f_X(x)}{f_X(x)} = \nabla_x \ln f_X(x), \tag{14}$$

where $f_X(x)$ is the pdf of the stochastic variable X, was used. This estimate was originally proposed by [12]. The denominator in the left hand side of equation(14) was estimated using equation(5) with a bandwidth, h = 0.05 and a "Normal Kernel", $\phi(x)$, defined as

$$\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}. \tag{15}$$

The Normal Kernel is equivalent to the pdf of a standard distribution with zero mean and unit variance. The numerator was estimated using equation(5) with the Kernel

$$K(x) = \frac{-x}{h}\phi(x), \tag{16}$$

where K(x) is the derivative of $\phi(x)$ with an additional factor $\frac{1}{h}$ that comes from differentiating the term $\phi(((x - x_i)/h)$ which shows up when K(x) from equation(16) is used in equation(5). The gradient operator is linear, it is therefore possible to differentiate before applying equation(5). The scores was estimated by combining equation(15) and equation(16).

E. Estimating H_1 and H_2

Equation(10) requires computation of filters $\overline{H_1}$ and $\overline{H_2}$, which model certain properties of the scanned papers.$\overline{H_1}$ and $\overline{H_2}$ were estimated using an adaptive filter algorithm. A common application for adaptive filters is to minimize functions. In this case, the goal was to minimize the error functions e_1 and e_2,

$$\begin{aligned}
e_1^{(t)}(m,n) &= f_r^s(m,n) \\
&+ \sum_{k=-K}^{K} \sum_{l=-L}^{L} l_1 \overline{H_1^{(t)}}(k,l) \times y_2^{(t)}(m-k, n-l) \\
e_2^{(t)}(m,n) &= f_v^s(m,n) \\
&+ \sum_{k=-K}^{K} \sum_{l=-L}^{L} l_2 \overline{H_2^{(t)}}(k,l) \times y_1^{(t)}(m-k, n-l),
\end{aligned} \tag{17}$$

where $2K + 1$ and $2L + 1$ is the size of the filters. This was accomplished using the Least Mean Square adaptive filter solution

$$\begin{aligned}
\overline{H_1^{(t+1)}} &= \overline{H_1^{(t)}} + \mu_{adaptive} f_r^s(m-k, n-l) e_1^{(t)}(m,n), \\
\overline{H_2^{(t+1)}} &= \overline{H_2^{(t)}} + \mu_{adaptive} f_v^s(m-k, n-l) e_2^{(t)}(m,n),
\end{aligned} \tag{18}$$

this operation was performed for k=-K...K and l=-L...L.

F. Motivating the choice of cost functions

By invoking equation(9) and neglecting q_i the following expression for the scanned recto image

$$f_r^s = f_r^i - l_1(f_v^i * \overline{H_1}) \tag{19}$$

can be obtained. Using equation(17), and letting $\widehat{H_1}$ be the estimate of $\overline{H_1}$, yields

$$e_1^{(t)}(m,n) = f_r^s(m,n) + l_1\widehat{H_1}(k,l) * y_2^{(t)}(m,n). \quad (20)$$

Replacing the term f_r^s in equation(20) by the expression developed in equation(19) yields

$$\begin{aligned} e_1^{(t)}(m,n) &= f_r^i(m,n) - l_1\overline{H_1}(k,l) * f_v^i(m,n) \\ &+ l_1\widehat{H_1}(k,l) * y_2^{(t)}(m,n). \end{aligned} \quad (21)$$

By linearity of the 2D-convolution operator and using the fact that as t increases, $y_2^{(t)}$ will approach f_v^i

$$e_1^{(t)}(m,n) \simeq f_r^i + l_1 y_2^{(t)}(m,n) * (\widehat{H_1} - \overline{H_1}). \quad (22)$$

Equation(22) obviously has a minimum when $\widehat{H_1} = \overline{H_1}$ which implies that the choice of $\widehat{H_1}$ that minimize e_1 is the choice that most closely resembles the real unknown $\overline{H_1}$.

In all equations used in this section, the iteration index t has been omitted from the filters for the sake of readability.

G. The ISJ4 algorithm

The algorithm requires images to be registered. Image registration is the process of aligning images in such a way that they are properly aligned in a single coordinate system. Consider, for example, a two-sided document with a stain visible on the top right corner of the front side. When the back side is scanned the show-through from the stain will typically appear in the top left corner instead because the page has been flipped. Registration can be used to align the images in such a way that the pixel pair (m,n) marks the start of the stain for both the back and front image. Two of the images used were preregistered but Fig. 5 was manually registered by selecting a few (4 or 5) control points (areas common to both images) and using them to algin the images.

After registration, the mean of the entire images were computed and subtracted from the images, resulting in zero mean inputs to the algorithm. The initial value of the components of the parameter vector and the coeffecients of the filters were all set to 0. The algorithm used to find the ML-estimate of $\overline{p}$ was

1) For all pixels (m,n), Compute $e_1(m,n)$ and $e_2(m,n)$ using equation(17).
2) Update the filters in all pixels using equation(18).
3) Use equation(10) to update $y_1^{(t)}$ and $y_2^{(t)}$.
4) Map the pixel intensities, in the images to $[0,1]$.
5) Update the parameter vector using equation(11) and the updated $y_i^{(t)}$.
6) Normalise $y_i^{(t)}$ again by subtracting the mean.
7) if convergence hasn't been achieved, go back to step 1.

When convergence had been achieved, the original DC-levels of the images was restored by adding the original means back to $y_1^{(t)}$ and $y_2^{(t)}$.

H. Convergence of the algorithm and sizing the filters

Two conditions were used in order to decide when convergence of the gradient ascent algorithm had been achieved. The algorithm was deemed to have converged when the difference between the euclidean norm of p_{i+1} and p_i was less than some threshold, T, of the euclidean norm of p_{i+1}, i.e.

$$\frac{N_e[p_{i+1} - p_i]}{N_e[p_{i+1}]} < T, \quad (23)$$

where N_e is the Euclidian norm operator. Alternatively, the algorithm was deemed to have converged when it had iterated a certain number of times.

Starting off, 3x3 size adaptive filters were used along with a 50 iteration limit and a threshold, T, of 3 percent. If satisfactory results were not achieved, for example if certain areas had not been removed properly, bigger sized filters might be needed. Increasing filter size sometimes led to extremely blurry images, in those cases a lower iteration cap of 30 or 40 might be sufficient. If convergence was achieved too fast, the threshold for convergence could be lowered to 2 percent. A threshold of 1 percent led to divergence in most cases.

All images used were grayscale images (converted from RGB in some cases) that were mapped to the interval [0, 1], i.e. pixel intensities took values from 0 to 1. After this mapping the standard deviation of the various images used were significantly smaller than 1, typically close to 0.17. As mentioned in the description of the separation algorithm, the mean of the scanned images were subtracted from each pixel resulting in images with a mean of zero. This procedure was necessary to avoid divergence of the adaptive filters. [2] recommends normalising the data in such a way that scanned images, used as inputs to the filters, have mean equal to 0 and standard deviation equal to 1. That approach made it significantly more difficult to make the algorithm converge and therefore the standard deviation of roughly 0.17 was kept intact for all images.

The mapping of pixel values to the interval [0,1], before computing the gradient, was not suggested by [2]. It was implemented because the gradient ascent algorithm typically diverged when this wasn't done.

[10] suggests applying a negative sign to the scores used in equation(13). This was tried, but frequently led to divergence of the gradient ascent algorithm or other unfavourable results. Therefore, no negative sign was applied to the score-function in the implementation described in this paper.

I. Results ISJ4

Applying the algorithm to Fig. 3 and Fig. 4 resulted in Fig. 18 and Fig. 19 respectively. Applying the algorithm to Fig. 1 and Fig. 2 using 3x3 sized adaptive filters, $\mu_{adaptive} = 8\times10^{-6}$ and $\mu_{ML} = 9\times10^{-3}$ and running 29 iterations of the algorithm resulted in Fig. 20 and Fig. 21, In both these cases, show-through was successfully removed, but white residue was left in its place. Applying regular thresholding easily solves this problem, as can be seen in Fig. 22 and Fig. 23. The algorithm was also run on a different segment of Fig. 5 and its respective verso side. The algorithm ran 50 iterations using

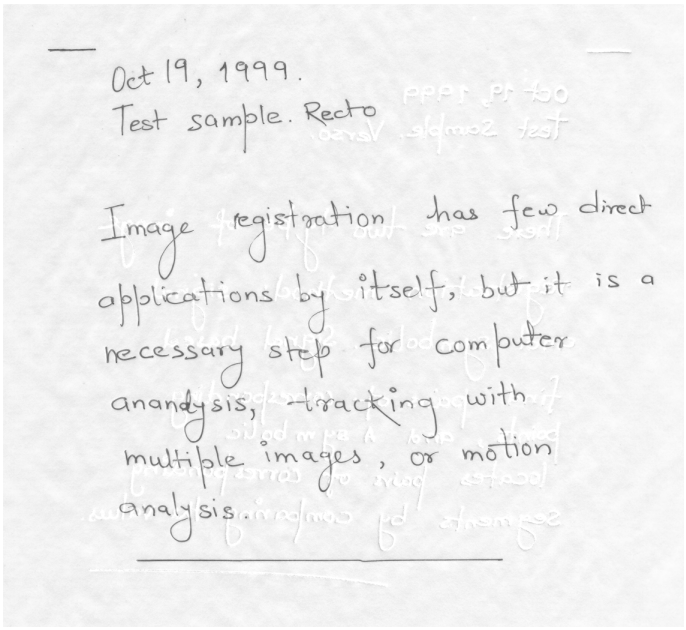


Fig. 18: ISJ4-algorithm applied to Fig. 3

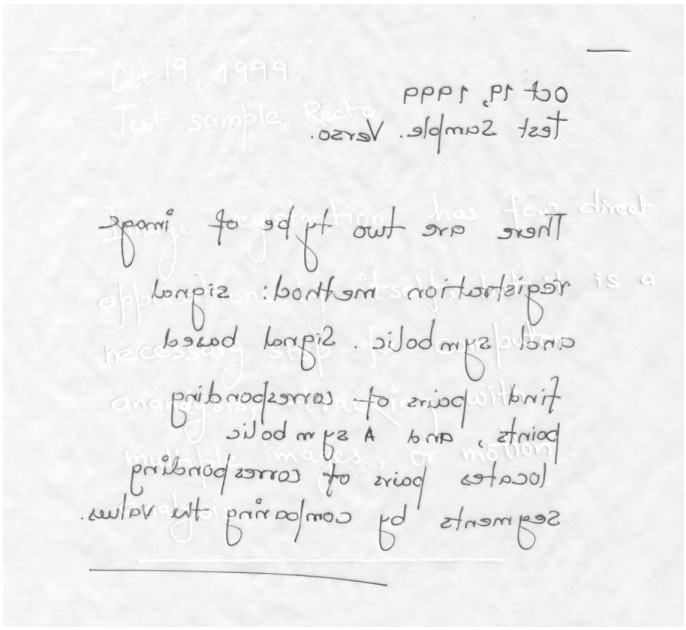


Fig. 19: ISJ4-algorithm applied to Fig. 4

3x3 filters, $\mu_{adaptive} = 1 \times 10^{-5}$ and $\mu_{ML} = 1 \times 10^{-3}$, which resulted in Fig. 24 and Fig. 25. In all cases presented the magnitude of the linear contributions, l_i were greater than the magnitude of the quadratic contributions, which is consistent with the theory developed by [2]. Typically the order of l_i was 10^{-1}, while the order of q_i was 10^{-2}. Furthermore, the linear factors, l_i had a negative sign, while the quadratic factors were positive.

J. Discussion

As can be seen in Fig. 24, the algorithm fails to remove all bleed-through. It is possible that if equation(7) were to be modified to include additional terms of the Taylor Expansion,

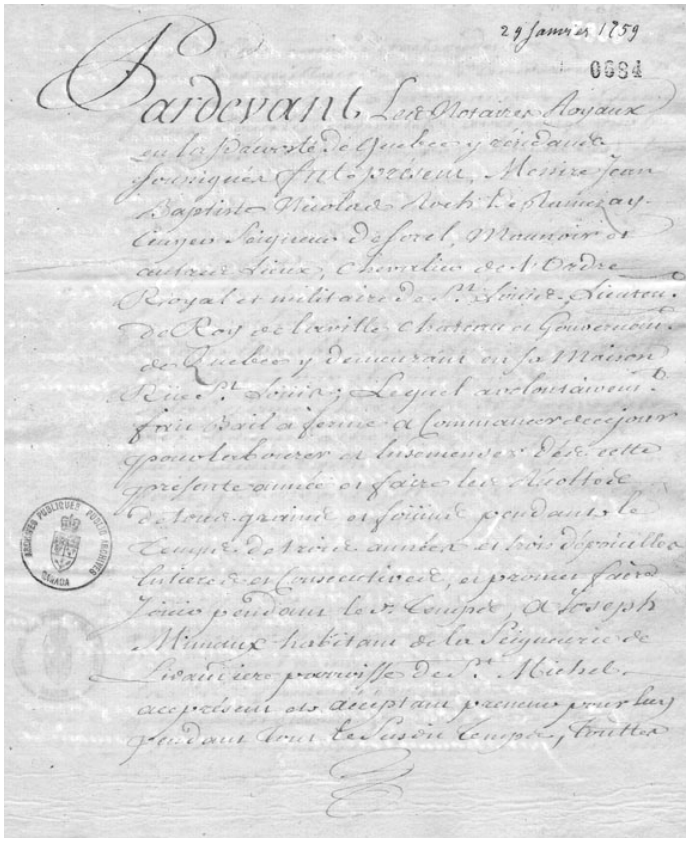


Fig. 20: ISJ4 applied to Fig. 1

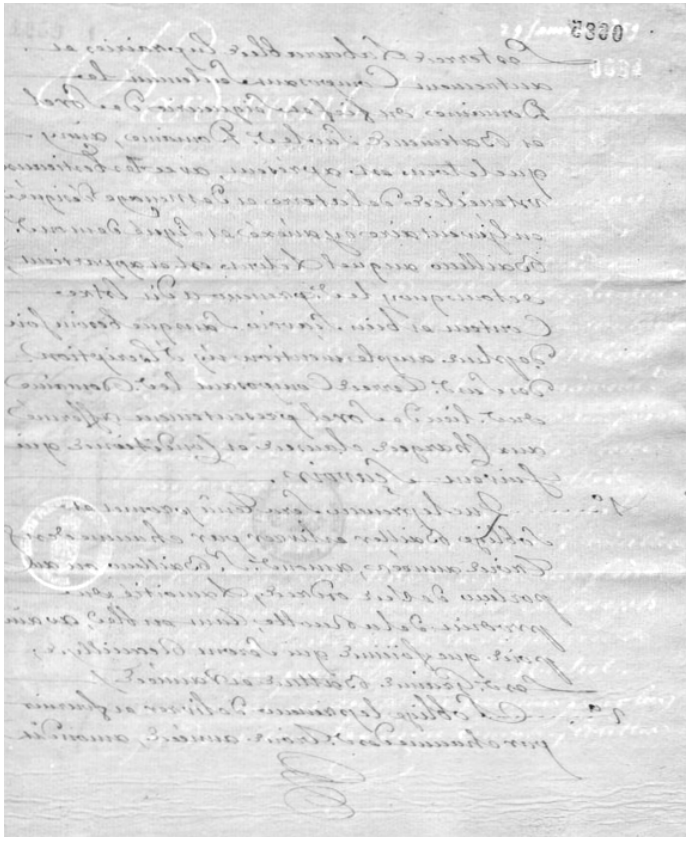


Fig. 21: ISJ4 applied to Fig. 2

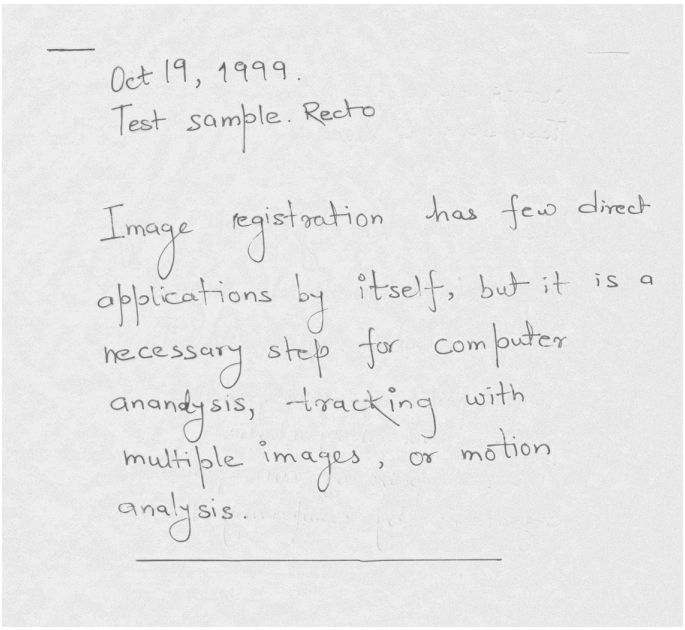


Fig. 22: Thresholded version of fig 18.

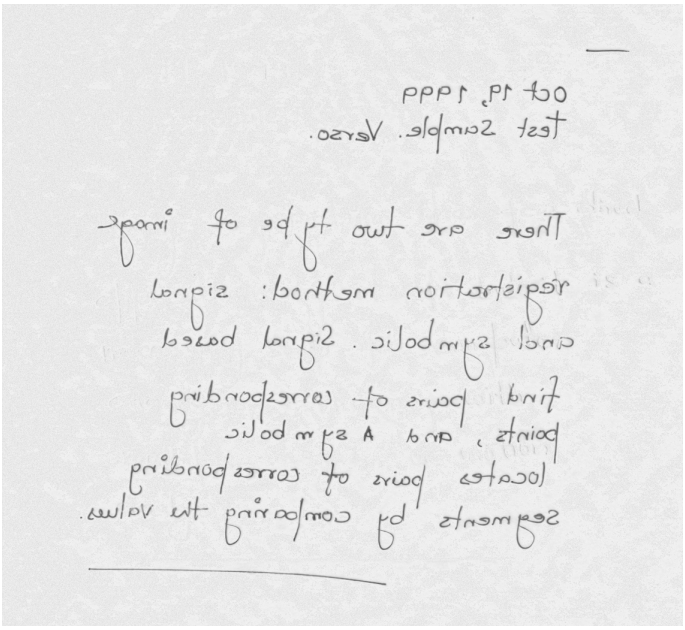


Fig. 23: Thresholded version of fig 19.

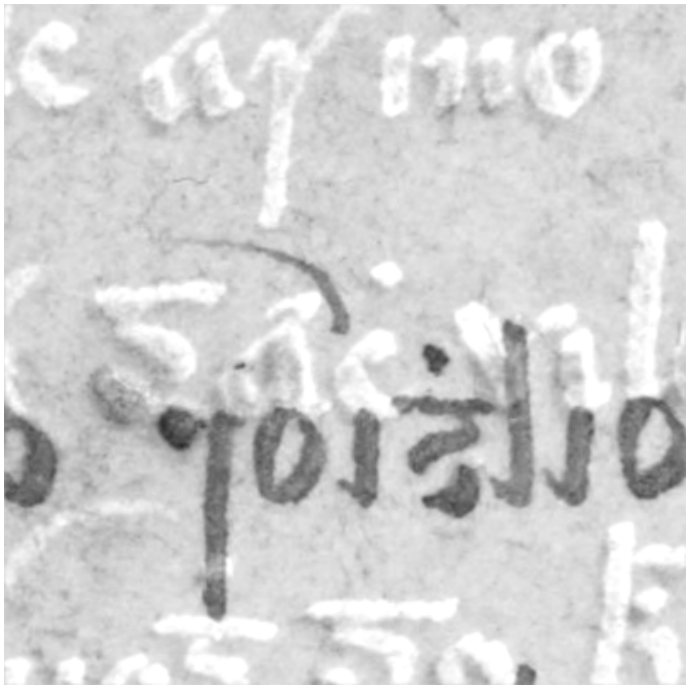


Fig. 24: ISJ4 applied to different segment of Fig. 5

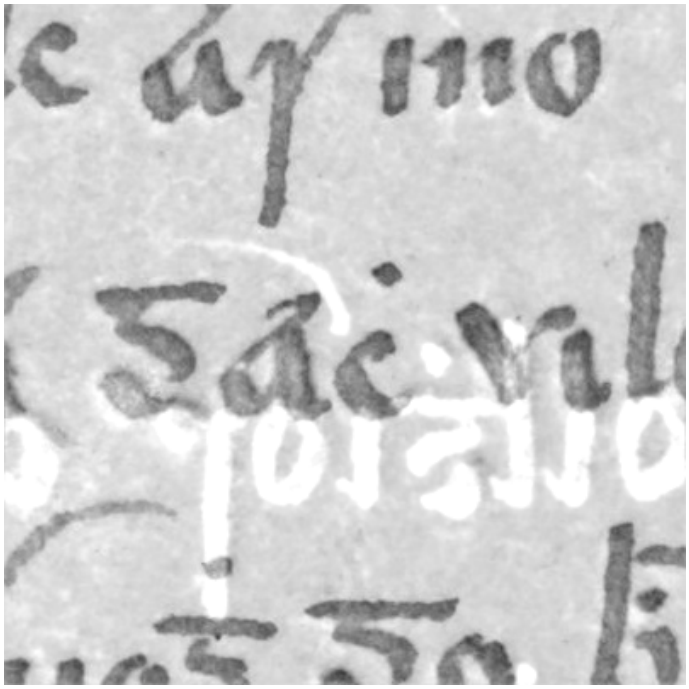


Fig. 25: ISJ4 applied to different segment of Fig. 5 respective verso side.

the algorithm would work better. Another possibility is that the algorithm is ill suited in general to deal with bleed-through because it was developed to take care of show-through.

Finding the right combination of learning rates that avoid divergence of both the gradient ascent algorithm, as well as the adaptive filters, can be time consuming. The images considered in this paper had somewhere in the neighbourhood of 10^5 to 10^7 pixels, and both the gradient ascent algorithm, as well as the updating of the filters, involve operations on every single pixel, in each iteration. One way to speed up the process of finding the right learning rates is to work with a smaller segment of the desired images. When a combination of rates that doesn't lead to divergence has been found, only then is the algorithm applied to the full size images used.

It is unclear if the implementation suggested in this paper is consistent with the way [2] intended. There are a couple of things that suggest that there might be unintentional differences between our implementations. One of these things is that we had massive problems making the algorithm converge using a normalisation with a standard deviation of 1. Another thing that might differ is that we used scores calculated without

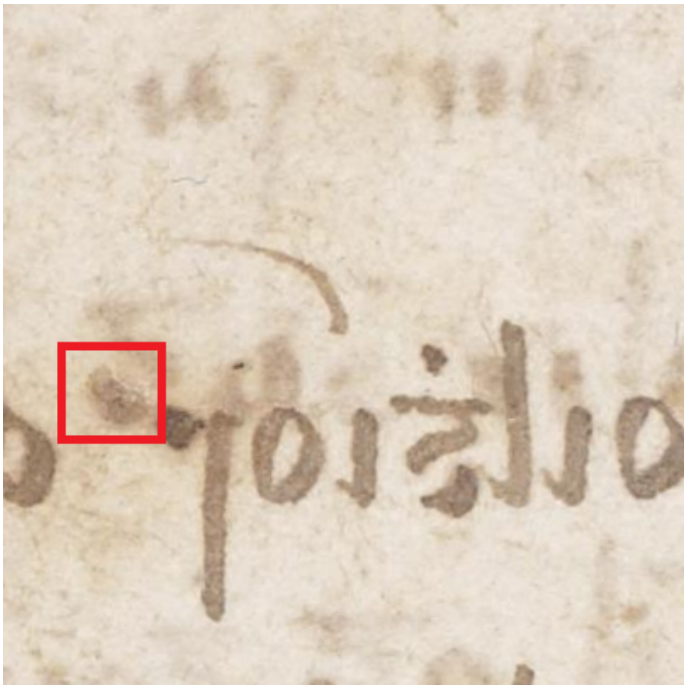


Fig. 26: The area inside the red rectangle represents bleed-through with high intensity.

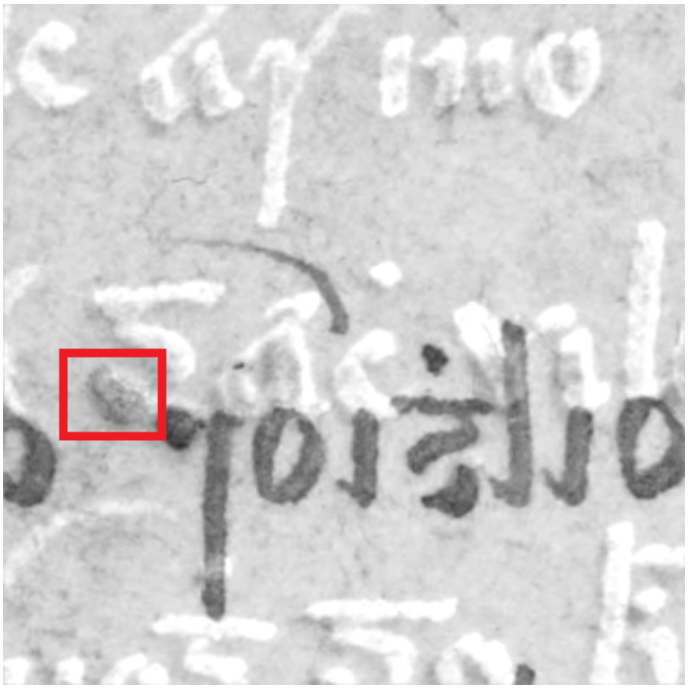


Fig. 27: The ISJ4 algorithm fails to fully remove bleed-through in the red rectangle.

a factor -1, as suggested by [10]. It is unclear if [2] used this factor or not, their implementation is not, in our opinion, easy to follow.

Overall this algorithm successfully removes most show-through, and various degrees of bleed-through, and thereby greatly improves readability. The residual white patterns in the images can easily be removed with Thresholding if needed. Allthough, some problems occured with bleed-through that was very heavy and irregular.

V. Significant bleed-through

In severe cases of bleed-through, such as in Fig. 26, the intensity of the bled ink is comparable to the intensity of the text on the affected side. In this particular case, the mixing model is not close to linear and ICA will therefore yield unfavourable results. ISJ4 is also not suitable for this type of problem which can be seen in Fig. 27 as equation(7) is derived under the assumption that the mixing model is almost linear. In order to address this problem, a segmentation technique based on Maximally stable extremal regions (MSER) was used in combination with ISJ4. MSER can be used to identify connected regions in an image based on certain mathematical properties. For details regarding MSER see [13]. The problematic area in Fig. 27 was located, using MSER-segmentation, and the pixels were set to white which resulted in Fig. 28.

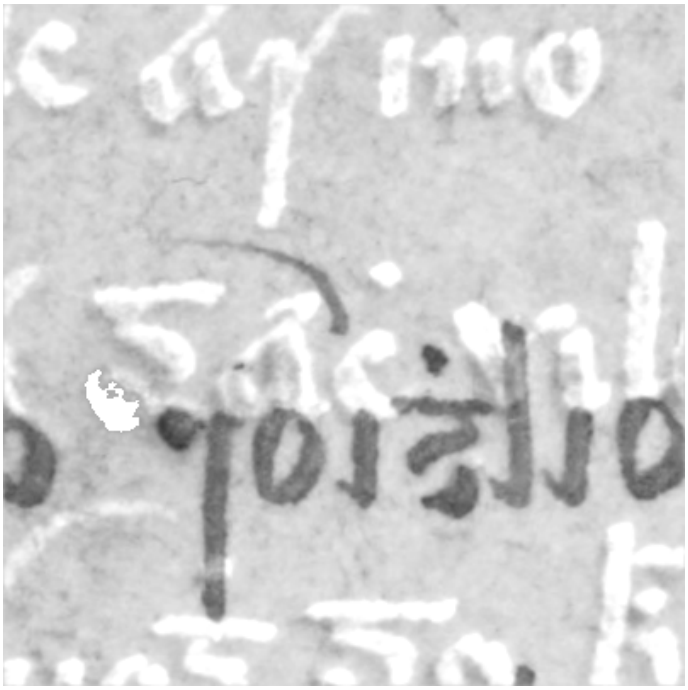


Fig. 28: Problematic area removed using MSER

VI. Independent Component Analysis

A. Theory

Independent component analysis (ICA) is a statistical technique for decomposing linear mixtures into independant sub-components. ICA is a way to perform blind source separation (BSS). The BSS problem can be stated as

$$x = As \tag{24}$$

where s is a vector containing the independent source signals, A is the unknown mixing matrix and x is the observed signals. If the mixing matrix A can be estimated, it's then possible to solve $s = A^{-1}x$. Three assumptions have to hold, for ICA to work properly;

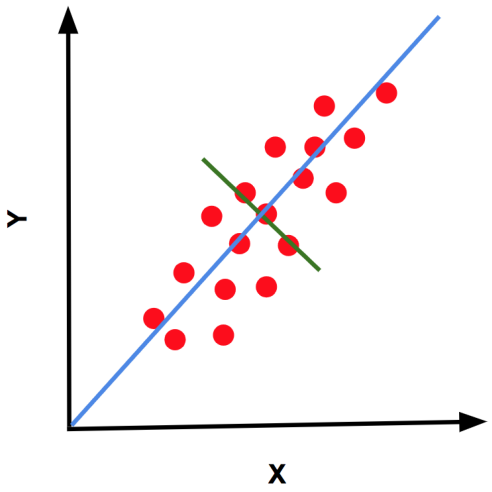


Fig. 29: PCA seeks to maximise data variance. Blue line represents the principal component of the data while the green line represents its second principal component. Source: Adapted from [14, Fig. 3]

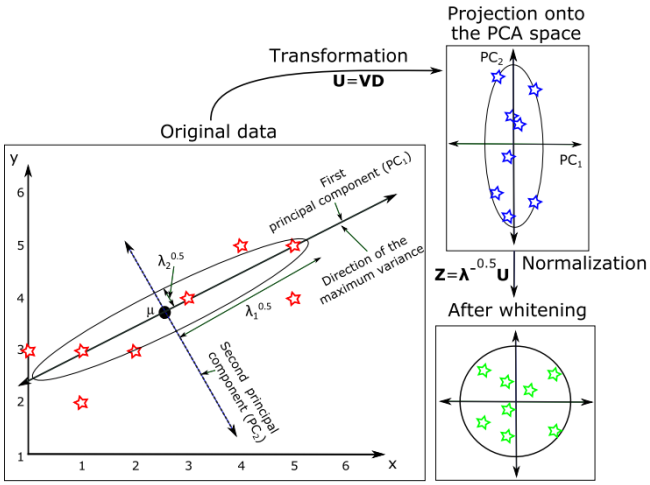


Fig. 30: Process of whitening the data. The centered data (Original Data in the figure) is first projected onto the PCA space of the data, then normalized into the whitened data. Source: Adapted from [15, Fig. 10]

1) The mixing Matrix A, is linear
2) The Source signals are Non-Gaussian
3) The Source signals are statistically independent

The preprocessing phase of ICA consists of centering, and whitening, the data. It is an important step because it simplifies many algorithms and reduces the number of parameters to be estimated. To center data means subtracting the mean from all the signals, $x - \mu$. To whiten data is to transform the signals to uncorrelated signals and then rescale each signal to have a unit variance, this can be done with the help of Principal Component Analysis (PCA). PCA is a dimensionality reduction method used for transforming high-dimensional data into lower-dimensional data. This could for example mean turning matrix data into vector form, or in general reduce dimensionality of matrix data. Dimensionality reduction is achieved by projecting data points onto the principal components.

PCA classifies the maximum variance, where the first principal component is that component which describes the maximum variance of the data. The second principal component is perpendicular to the first principal component and it too is in the direction of maximum variance, see Fig. 29.

It turns out the principal components are eigenvectors of the data's covariance matrix, and the eigenvector with the largest corresponding eigenvalue is the first principal component. PCA can be used to find the eigenvectors and their corresponding eigenvalues of the data's covariance matrix.

Two random variables that are uncorrelated must have a covariance of zero. This can be accomplished through the use of transformations.

The centered signals are projected onto the PCA space, $U = VD$, where V is the eigenvectors of the centered data's covariance matrix and D is the centered data. This makes the signals decorrelated, meaning their covariance is zero. In order to obtain unit variance for each decorrelated signal, the matrix U, containing the decorrelated signals, is multiplied by $\lambda^{-\frac{1}{2}}$, where λ is the eigenvalues of the data's covariance matrix, and $\lambda^{-\frac{1}{2}}$ is defined as the componentwise operation $\lambda^{-\frac{1}{2}} = [\lambda_1^{-\frac{1}{2}}, \lambda_2^{-\frac{1}{2}}, ..., \lambda_n^{-\frac{1}{2}}]$. The resulting matrix $Z = \lambda^{-\frac{1}{2}}U = \lambda^{-\frac{1}{2}}VD$ is a matrix containing the preprocessed data. The covariance matrix of Z should satisfy the relation $\mathbf{E}\{ZZ^T\} = I$, where I is the identity matrix. This relationship shows that the covariance between the signals is zero and that each signal has unit variance. The process of whitening the data is summarised in Fig. 30. The next step is to estimate the unmixing matrix W, where $W = A^{-1}$. Finding an exact solution for W is not possible because the A matrix is unknown. Estimating W can be done in several ways. A popular choice is maximizing the non-Gaussianity of extracted signals through the use of Kurtosis. Non-gaussianity is a key concept in ICA. It makes use of the Central Limit Theorem which says that a sum of independent random variables tends towards a gaussian distribution. It is therefore the case that a linear combination of two independent random variables is usually more gaussian than either of the two independent random variables alone. On account of this, the unmixing matrix W should maximise the non-Gaussianity of the extracted signals, in order to make the extracted signals as independent as possible.

To be able to maximise non-Gaussianity, non-Gaussanity has to be measurable. Kurtosis can be used as a measure of gaussianity. It is defined as the fourth standardized moment, where

$$K(X) = \frac{E[(X-u)^4]}{(E[(X-u)^2])^2} - 3 \tag{25}$$

is the kurtosis of X. The absolute value of the Kurtosis relates to the shape of the distribution. The higher the absolute value is, the less Gaussian the distribution is. It's worth noting that

Gaussian pdfs have zero kurtosis after including the -3 term. Let $Q = A^T W = A^{-1} W$, now let $y = W^T x = W^T As = Qs = q_1 s_1 + q_2 s_2$.

When the absolute value of the kurtosis of y is maximised, the unmixing matrix will have extracted one source signal. y will be equal to one of the source signals because one of the q terms will be zero. In order to find all the independent components, the local maxima of the kurtosis of y are located. A way to minimise the kurtosis of y is through the use of gradient ascent, which finds the local maximas.

B. Implementation

The assumption about having a Linear mixing matrix A does not hold, since the processes of bleed-through (ink diffusion) and paper absorption is non-linear. This can be seen in equation (6). If the nonlinear contributions are neglected, equation (6) simplifies into a linear one [15]. Consider $r(t)$ and $v(t)$, the grayscale scanned recto and verso side respectively. Let $r(t)$ and $v(t)$ be a linear combination of the two independent source images s_1 and s_2. Here $t = 1, 2, ..., M \times N$, where M and N is the rows and columns, respectively, of the scanned images.

$$\begin{aligned} r(t) &= A_{11} s_1(t) + A_{12} s_2(t) \\ v(t) &= A_{21} s_1(t) + A_{22} s_2(t) \end{aligned} \tag{26}$$

Here equation(26) is a simplified model, consisting of linear combinations of the source signals.
Assumption 1 holds, after the simplification.
Assumption 2 (Source signals are Non-Gaussian) holds, since the intensity levels of written text is not gaussian in nature.
Assumption 3 holds, since the text on either side can be expected to be independent from each other. For mixtures where all assumptions hold, ICA is a good method for separating s_1 and s_2. For all images, equation 26 was assumed to hold. The ICA algorithm was then applied in accordance with the theory section.

C. Results

Fig. 31 shows two pieces of a bleed-through document, before and after ICA has been applied. It shows that a lot of bleed-through has been removed, though in the top right the cancellation is visible. The original document shows consistent bleed-through. Fig. 32 shows the results ICA has on a realtively linear mixture. It removes show-through convincingly. Left side of Fig. 33 shows signs of irregular, spotty bleed-through. Right side shows the changes after ICA has been applied. In this case the changes were not noticeable.

D. Discussion

ICA works relatively well for removing show/bleed-through in most documents. The simplification seems not to matter in most cases. However, in some cases where there exists heavy, non-constant bleed-through, as in Fig. 33, the ICA method didn't show a good result. The image was practically unaltered through the ICA process.
ICA works well for show/bleed-through documents where the linear simplification holds, but has problems with patchy and uneven bleed-through.

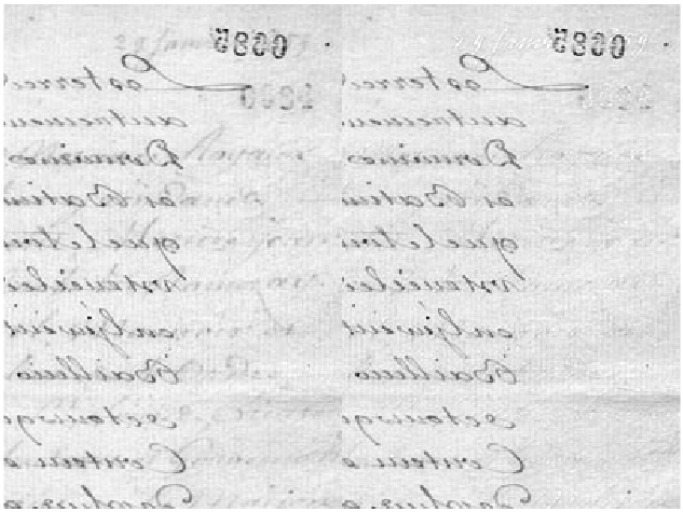


Fig. 31: Montage of upper left side of original document vs after ICA has been applied

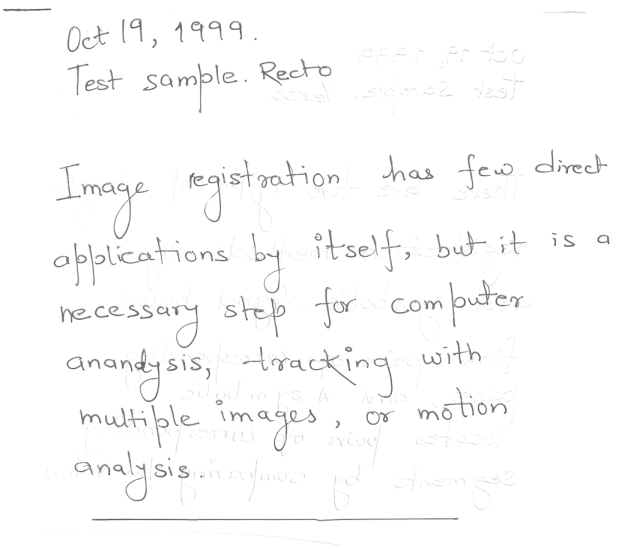


Fig. 32: ICA method having been applied to a show-through document

VII. Discussion

A. Comparing the algorithms

One option to handle really strong and irregular bleed-through Fig. 5 was ISJ4 in combination with MSER segmentation. ICA was not useful for this type of image with varying degrees of ink bleed, because the linearity assumption doesn't hold for this type of image.

MSER segmentation was used successfully in combination with ISJ4 in a case where most of the bleed-through could be addressed by ISJ4, but the algorithm failed to detect an isolated case of really strong bleed-through. Using only MSER segmentation to address all show-through and / or bleed-through in an image would be impractical as text results in a large number of segmented areas that need to be processed.

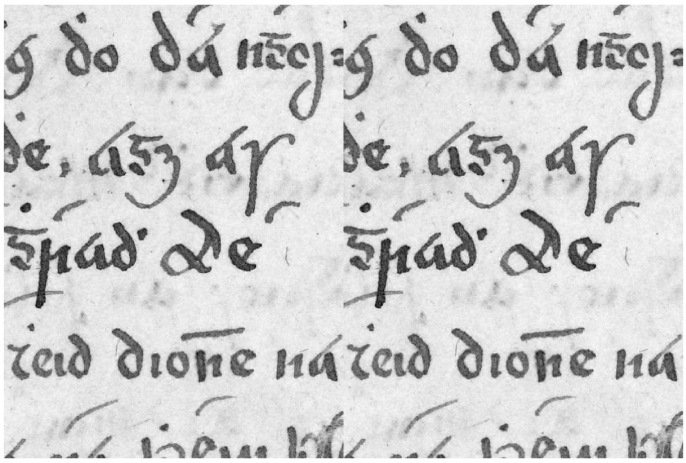


Fig. 33: Left shows the original piece of document, right side shows the same piece after ICA has been applied

Furthermore, it is difficult to detect all text in one go, so multiple runs using different sensitivity or different segmentation techniques would be needed. It follows that text segmentation is perhaps best used in combination with different algorithms, when needed, and not as a primary way to remove bleed-through or show-through. Removing show-through, or bleed-through, using text segmentation requires a lot of manual work and is therefore not suitable for cleaning up large amounts of texts.

ISJ4 as well as ICA yielded excellent results with more linear mixtures. The benefits of using ICA over ISJ4 are primarily speed. ISJ4 can be computationally time-consuming (depending on image-size) and it takes time to figure out the right combination of learning rates and filter size. Execution of ICA is fast and requires no tuning of parameters. While Fig. 5 was difficult to separate for both ICA and ISJ4, ISJ4 managed to remove most, but not all, bleed-through.

Thresholding was useful for images when the show/bleed-through is distinct in pixel intensity from the foreground. If this is the case then the foreground and the bleed or show-through can easily be separated. This holds even when the show/bleed-through is uneven, or differs locally, which is a problem for many of the other algorithms studied. This makes the method suitable for documents with bleed-through that are not as dark as the foreground. Thresholding was also useful for removing residual effects obtained using ISJ4.

The Dubojs algorithm makes no assumption about linear mixtures and was therefore better than ICA at handling non-linearities. The algorithm shows no benefits compared to ISJ4 with regards the 'nice' mixtures such as Fig. 3 and Fig. 1. The reason for this is because the segmentation method was not doing a good enough job of separating R2 and R4 regions. A perfect classification of R2 and R4 would yield excellent results, but is not possible with the method described in [5]. This is because Dubojs is a more sophisticated thresholding method. A value is chosen to use as threshold for correlation, or contrast comparisons. Choosing a thresholding value is always a compromise as some pixels inevitably gets misclassified in the process.

Dubojs yielded reasonably results good with the difficult case of case irregular and heavy bleed-through i.e. Fig. 5. This heavy bleed-through is caused by really thick lettering in the handwriting, this in turn is why Dubojs is useful. As previously mentioned, Dubojs will misclassify some pixels, resulting in unintentional inpainting of some foreground pixels. The thicker the foreground is, the less of a problem this misclassification is. Shaving a few pixels off of really thick lettering was barely noticeable.

VIII. Conclusions

It was shown that choice of method for show-through or bleed-through removal is heavily dependent on type of document. If the mixtures was uncomplicated enough, i.e. degree of bleed-through was consistent across the document and not comparable in intensity with foreground, choice of method wasn't important. In this case, all methods considered gave reasonable results. A specific type of bleed-through document with very thick lettering and subsequent strong bleed-through caused problems. Two of the method's considered yielded reasonable results in this case but further research is needed to handle this case.

A short summary of methods considered and their usefulness can be found in table I. In this table, usefulness is graded on a scale 0 to 2, where 0 represents not useful, 1 represents somewhat useful, and 2 represents really useful.

TABLE I: Final conclusions summarised

Method considered	Show-through	Bleed-through	Heavy bleed-through	irregular bleed-through	irregular heavy bleed-through
Thresh	2	2	0	0	0
ICA	2	2	2	0	0
Dubojs	2	2	2	2	1
ISJ4	2	2	2	2	0
MSER	2	2	2	2	2

A. Future work

Further research could attempt to update the ISJ4-model to make it better suited at difficult mixtures, such as very heavy ink-bleeding. The Dubojs algorithm can potentially be improved by using different segmentation methods. Another interesting approach would be to look at look closer into the specific type of heavy bleed-through that caused problems for the algorithms considered in this paper, as opposed to treating bleed-through as one class of problems. Another area of interest would be to evaluate the efficiency of Machine Learning methods and compare them to the methods described in this paper. Using Machine Learning methods to solve the problem of blindly mixed images has been done. One could implement them and evaluate their complexity and efficiency, relative to the methods described in this paper.

References

[1] G. Sharma, "Show-through cancellation in scans of duplex printed documents," *IEEE Transactions on Image Processing*, vol. 10, no. 5, pp. 736–754, Dec 2001.

[2] F. Merrikh-Bayat, M. Babaie-Zadeh, and C. Jutten, "Linear-quadratic blind source separating structure for removing show-through in scanned documents," *IJDAR*, vol. 14, pp. 319–333, Dec 2011.

[3] A. Tonazzini, E. Salerno, and L. Bedini, "Fast correction of bleed-through distortion in grayscale documents by a blind source separation technique," *Document Analysis and Recognition*, vol. 10, pp. 17–25, Jun 2007.
[4] B. Ehsandoust, M. Babaie-Zadeh, B. Rivet, and C. Jutten, "Blind source separation in nonlinear mixtures: separability and a basic algorithm," *IEEE Transactions on Signal Processing*, vol. 65, no. 16, pp. 4339–4352, Jul 2017.
[5] E. Dubois and P. Dano, "Joint compression and restoration of documents with bleed-through," Apr 2005, pp. 170–174.
[6] Dublin Institute for Advanced Studies. Irish script on screen database. Ireland, Dublin 4, 10 Burlington Road. [Online]. Available: https://www.isos.dias.ie/english/index2.html
[7] Dubois, Eric. (2011, Sep.) Index of / edubois/documents. [Online]. Available: https://www.site.uottawa.ca/ edubois/documents/
[8] M. Sonka, V. Hlavac, and R. Boyle, *Image processing, analysis, and machine vision*, 4th ed. Cengage Learning, Jan 2014.
[9] Murzova, Anastasia and Sakshi Seth. (2020, Aug.) Image thresholding. Learnopencv, San Diego, CA, USA. 2021, Mar. [Online]. Available: https://learnopencv.com/otsu-thresholding-with-opencv/
[10] S. Hosseini and Y. Deville, "Blind maximum likelihood separation of a linear-quadratic mixture," in *International Conference on Independent Component Analysis and Signal Separation*, Oct 2004, pp. 694–701.
[11] ——, "Correction to:" blind maximum likelihood separation of a linear-quadratic mixture"," *arXiv preprint arXiv:1001.0863*, Jan 2010.
[12] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on information theory*, vol. 21, no. 1, pp. 32–40, Jan 1975.
[13] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, vol. 22, no. 10, pp. 761–767, Sep 2004.
[14] Mallick, Satya. (2018, Jan.) Principal component analysis. Learnopencv, San Diego, CA, USA. [Online]. Available: https://learnopencv.com/principal-component-analysis/
[15] A. Tharwat, "Independent component analysis: an introduction," *Applied Computing and Informatics*, vol. ahead-of-print, p. 15, Aug 2018.

Machine Learning-based Biometric Identification

Hanna Israelsson and Andreas Wrife

***Abstract*—With the rapid development of computers and models for machine learning, image recognition has, in recent years, become widespread in various areas. In this report, image recognition is discussed in relation to biometric identification using fingerprint images. The aim is to investigate how well a biometric identification model can be trained with an extended dataset, which resulted from rotating and shifting the images in the original dataset consisting of very few images. Furthermore, it is investigated how the accuracy of this single-stage model differs from the accuracy of a model with two-stage identification. We chose Random Forest (RF) as the machine learning model and Scikit default values for the hyperparameters. We further included five-fold cross-validation in the training process. The performance of the trained machine learning model is evaluated with testing accuracy and confusion matrices. It was shown that the method for extending the dataset was successful. A greater number of images gave a greater accuracy in the predictions. Two-stage identification gave approximately the same accuracy as the single-stage method, but both methods would need to be tested on datasets with images from a greater number of individuals before any final conclusions can be drawn.**

***Sammanfattning*—Tack vare den snabba utvecklingen av datorer och modeller för maskininlärning har bildigenkänning de senaste åren fått stor spridning i samhället. I denna rapport behandlas bildigenkänning i relation till biometrisk identifiering i form av fingeravtrycksavläsning. Målet är att undersöka hur väl en modell för biometrisk identifiering kan tränas och testas på ett dataset med ursprungligen mycket få bilder, om datasettet först expanderas genom att flertalet kopior av originalbilderna skapas och sedan roteras och förskjuts i olika riktningar. Vidare undersöks hur noggrannheten för denna enstegsmodell skiljer sig jämfört med identifiering i två steg. Vi valde Random Forest (RF) som maskininlärningsmodell och Scikits standardinställningar för hyperparametrarna. Vidare inkluderades femfaldig korsvalidering i träningsprocessen. Prestandan hos den tränade maskininlärningsmodellen bedömdes med hjälp av testnoggrannhet och *confusion matriser*. Det visades sig att metoden för att expandera datasettet var framgångsrik. Ett större antal bilder gav större noggrannhet i förutsägelserna. Tvåstegsidentifiering gav ungefärligen samma noggrannhet som enstegsidentifiering, men metoderna skulle behöva testas på datamängder med bilder från ett större antal individer innan några slutgiltiga slutsatser kan dras.**

***Index Terms*—Machine Learning, Biometric identification, Classification, Dataset expansion, Random forest**

Supervisors: Tobias Oechtering and Linghui Zhou

TRITA number: TRITA-EECS-EX-2021:190

I. Introduction

Computer image recognition is a powerful tool currently adopted in many areas. Most commonly, image recognition is used in the healthcare, automotive and security industries to analyze and identify patterns and behaviors [1]. A concrete example can be found in modern medicine where practitioners, such as doctors and nurses, use ultrasound, CT and MRI scans on a daily basis. These images were previously manually analyzed, but are now analyzed by computers. The results support doctors and researchers in their aim to recognize various diseases and medical conditions, so that the correct remedy can be given to the patient. Another example of the use of image recognition is in the automotive industry, where self-driving cars use this type of technology to identify their surroundings. Cameras situated all-around a car give live feedback to a computer that analyses the road and potential obstacles [1].

Lastly, and most importantly to this thesis, is the use of image recognition in the security and protection sector [1]. Here, the main goal, oftentimes, is to identify a human being based on certain characteristics. This identification process is typically called biometric identification. In security applications, biometric identification can be used as passwords to grant access to systems such as computers and phones [1]. Moreover, in criminology, biometric identification is used to find the perpetrator.

In order to perform biometric identification, the various traits of a human being need to be distinguished. One can categorize traits by treating them as being either physical or behavioural. For instance, some examples of physical traits are DNA, irises, fingerprints and facial patterns. Furthermore, examples of behavioural traits include voice and handwriting. Once traits have been distinguished, values of each trait need to be obtained. This is typically done by dividing the image into smaller sections and extracting relevant information [2]. Then, once data has been obtained, a machine learning algorithm can be constructed and trained on the traits - which also are known as being the independent variables.

The goal is to design an algorithm that can classify an image with a high level of precision. To classify the biometric feature of individuals, we assign each individual with a label and we want to reliably guess the corresponding label with an observation, e.g. scanned fingerprint. The dependent variable can be a name, a social security number, or a group of people. To make the identification process more efficient, a technique known as the multi-stage identification process can be used [3]. This means first classifying a depicted object into a category and then later, now on a reduced dataset, try to identify it.

A. Main Project Objectives

In this thesis project we will investigate

1) how the machine learning process is affected by the use of a limited dataset.
2) if two-stage identification can be used with the machine learning algorithm to increase the identification accuracy, compared to a single-stage setup.

II. Background

A. Machine Learning

Machine learning is a crucial standpoint in Big Data and Artificial Intelligence (AI). It allows computer software to learn autonomously. The idea is to use mathematical algorithms to identify patterns, learn from them and thereafter make decisions and predictions. There are many sub-branches of machine learning, notably different algorithms applied to different problems. There are also different methods of training the algorithm. In this project, supervised random forest [4] is studied and used.

B. Supervised Learning

Problems in machine learning can be placed into two categories: supervised learning and unsupervised learning. Supervised learning means that the algorithm is trained with a preexisting dataset. The data consists of observations or predictions, labeled for the computer to understand. An example of a label on a fingerprint is on a left (0) or right (1) hand [5]. Unsupervised learning is the opposite of supervised, where the algorithm does not have access to any labeled data. The algorithm is instead trained using trial and error. This could for instance be a virtual car learning how to follow a track. To incentivise unsupervised learning, a goal is usually set. Other options include having clear right and wrongs. In the example mentioned this could be telling the car to start over if it goes off the road. In this project, the algorithms are trained using supervised learning. The aim is to fit a model that accurately predicts the label of a fingerprint similar to one already in the dataset. Understanding the relationship between the label and the predictors better.

C. Dataset

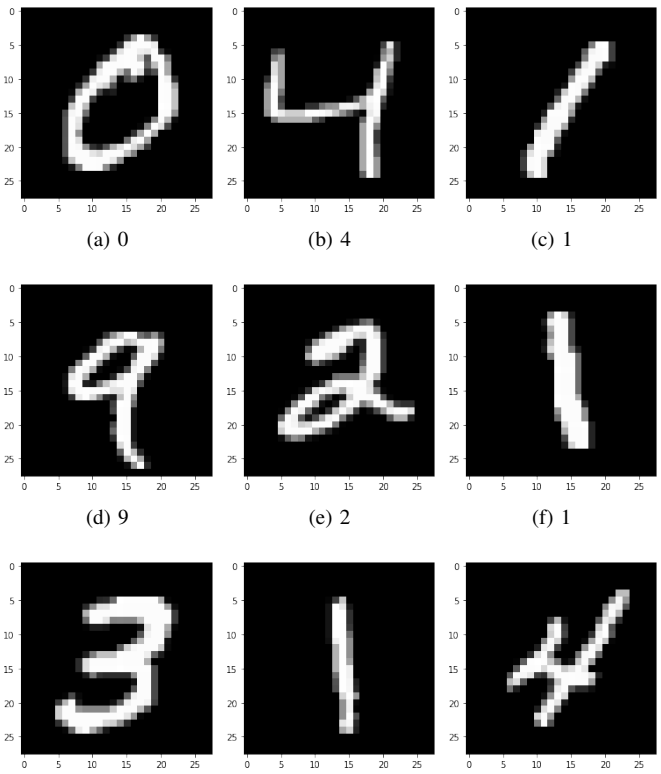
1) SOCOFing: For the algorithm to identify a user it needs to know the user beforehand, this is done by implementing a dataset. The dataset is a collection of a certain type of biometric. This could for instance be the user's face in different angles. For this project, a dataset of fingerprints will be used, which is the *Sokoto Coventry Fingerprint Dataset* (SOCOFing) [5]. This dataset consists of 6000 unique fingerprints from 600 people. Ten fingerprints for each person, one for each finger. All the fingerprint images are 97x90 pixels in size. Besides the identity behind the fingerprint, each fingerprint is categorized and assigned with labels. These include gender (male or female), hand (left or right) and digit (little, ring, middle, index, thumb). The images are labeled by the form *[identity]__[gender]_[hand]_[finger]*, for example 100__M_Left_index_finger , see Figure 1.

The set of images is also available in altered form, where each fingerprint has been slightly adjusted to incorporate the uncertainty between measurements. A clear example of this is if the individual had an injury and a scar is noticeable on the fingerprint. The modified fingerprints come in three difficulties: obliteration (easy), central rotation (medium), and z-cut (hard) [5], adding _difficulty to the end, see Figure 1. These altered forms allow the biometric identification algorithm to be prepared to recognize a fingerprint regardless of slight alterations. All fingerprints from the SOCOFing dataset are in black and white which translates to matrices with single values for each pixel ranging between 0 (black) and 255 (white).

(a) Left index finger (b) Left little finger (c) Left middle finger

Fig. 1. Three examples of images from the SOCOFing dataset [5]. All three fingerprints are from the same male of identity 100.

2) MNIST Dataset: Another dataset that was used, but to a lesser extent, was the MNIST dataset [6]. The MNIST dataset consists of 60000 images of handwritten digits, ranging between zero and nine. Each image contains a digit and has an image size of 28x28 pixels. This dataset was implemented for testing and comparison purposes. Implemented to evaluate the effectiveness of the different algorithms used and to see if they could be applied to other datasets. See Figure 2 for more detail.



(a) 0 (b) 4 (c) 1

(d) 9 (e) 2 (f) 1

(g) 3 (h) 1 (i) 4

Fig. 2. Nine examples of images from the MNIST dataset [6].

D. Dataset Expansion

Although the SOCOFing dataset [5] consists of a lot of images, including the alterations, each original fingerprint

is only enrolled once. To be able to train the supervised machine learning algorithm well, more slight alterations of each fingerprint is required. In other words, a way to extend the dataset is needed.

One such strategy for extending the dataset is to make multiple copies of each fingerprint and then rotate them a random number of degrees. In doing so, the values of most pixels in the images are preserved, only appearing at slightly different positions, see Figure 5. The neighboring pixels can also be kept, which allows the systematized process to identify patterns of recurring sets of pixels.

Another approach is to shift the entire image in a random direction (left, right, up or down) by a few pixels. This creates a new setup for the algorithm to decipher. The two approaches can also be combined to generate a two-set difference in imagery, to further increase the efficiency of the identification process.

E. Single-Stage Identification

The identification process will now be explained in more detail. In Figure 3, a simple image recognition system is depicted, adjusted for biometric identification.

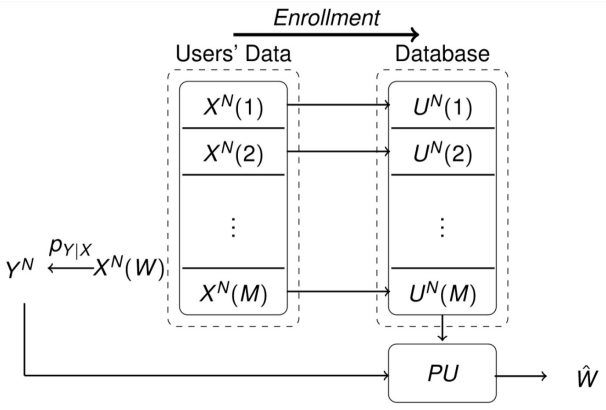


Fig. 3. A simple single-stage identification process. In the enrollment phase, the user's data, X^N, is enrolled in the database. In the identification phase, the processing unit, PU, compares the observation, Y^N, with the database to identify the user's identity.

As depicted in Figure 3, an identification system consists of two phases, i.e., the enrollment phase and the identification phase [7]. In the enrollment phase, the biometric data of users are enrolled into a database. In more detail, a model is trained with a selected algorithm such as Random Forest or a Neural Network. The biometric sequences $X^N(w)$ are stored as $U^N(w)$ in the dataset, which in this project includes the extension of the dataset, but exclude the easy, medium and hard altered copies. Part of these altered images is also the observation Y^N. Usually, these observations are part of a new dataset, captured with a camera or similar devices. Some of these observations are likely to include noise, such as a blurry image. With a trained model in the processing unit (PU), the observation is compared with the database to guess the user of the former. In other words, the observation Y^N is compared with the database U^N and identity W is predicted. As stated before, the aim of the algorithm is to reliably guess the correct user of an observation Y^N. An algorithm with a higher verification rate is a better biometric system. An example with few enrolled users is on an individual's phone, only the phone owner's face or fingerprint is enrolled.

F. Multi-Stage Identification

Another application of the machine learning method is to classify the users into categories. The more users in a database, the more complex the search becomes. The PU has more biometric sequences enrolled which prolong the process of identifying the correct user. This is the case in some real-life scenarios, for instance, a CCTV camera placed in a crowded area. To reduce the search complexity of the system, a hierarchical identification can be introduced [3]. This means, the identification process is divided into a multi-stage identification process, that first categorize users based on certain characteristics and labels and then subsequently identifies the user on the now smaller dataset.

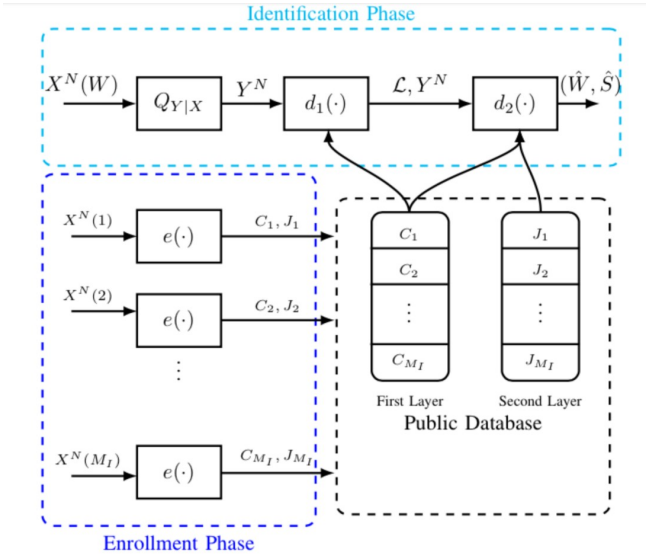


Fig. 4. The figure depicts a multi-stage identification process. In the enrollment phase, the user's data are enrolled in the dataset, consisting of helper data C_i and J_i. The helper data is then used to identify an observation Y^N.

A hierarchical biometric identification system is illustrated in Figure 4. In the enrollment phase, the users' biometric sequences, X^N, are mapped to the first layer helper data, C_i, and the second layer helper data, J_i, stored in a database. In the identification phase, similar to the basic setting, an unknown user, W, is observed. From this, an observation Y^N is obtained. Afterwards, the identification process commences. This is done in two stages. Firstly, the system compares the observation with the first layer helper data C_i in $d_1()$. The output is a list, $\mathcal{L}$, which contains a list of cluster indices used together with the second stage $d_2()$. Here, the algorithm compares the observation with the user only contained in the list $\mathcal{L}$. Thus, the complexity is reduced as the users, not in the same category are excluded.

There is a similarity between the biometric identification and classification, i.e., they are both interested in looking

for the relationship between the observation Y^N and the corresponding labels. In the case of the SOCOFing dataset, these includes the mentioned labels gender, hand and finger. Although there are many similarities between classification and identification, there is a difference between the required number of data samples. Identification requires only one enrollment for each user, whilst the classification demands more unique images for each category [7].

G. Machine Learning Models

Random Forest is an algorithm that uses decision trees to finish the classification task. The algorithm reduces the input into smaller branches and processes them individually to then later combine them to predict the users. In each decision tree, there are multiple three branches that split the data depending on if the data fits the criteria.

The tree have three hyperparameters, a choice of how many layers and what type of layers it consists of, random forest trees can have different sizes. Similar to determining the number of layers and the types of layers, three parameters have to be determined in the tree, which are node size, the number of trees, and the number of sampled features. In this project, only the number of trees was changed from the base case. This is further described in the method.

H. Evaluation of Results

The performance of the identification processes and dataset expansion has to be evaluated somehow. Methods for this evaluation, that are used in this report, are cross-validation and confusion matrices.

1) Cross Validation: Cross-validation is a method for generating more reliable test accuracies. Normally, when a machine learning algorithm is tested, one only receives the accuracy of one specific training/testing split. With the method of cross-validation, multiple different splits are used, and the mean of the accuracies of those can be used for the evaluation of the model. For example, if five-fold cross-validation is used, the dataset is split into five parts, each consisting of 20% of the full dataset. For instance, label those five parts A, B, C, D and E. In the first evaluation, A is used for testing and B, C, D and E are used for training. The second time, B is used for testing and A, C, D and E are used for training. When all parts have been used for testing, one time each, the mean of the five generated test accuracies can be used as a more credible measurement of the test accuracy of the model.

2) Confusion Matrices: A confusion matrix, also called an error matrix, is a table with predicted values on the x-axis and true values on the y-axis. See for example Figure 8. The number in each cell of the matrix represents the number of predictions of that specific sort. For example, if row 0 consists of the numbers $64, 12, 31, 9, 13, 17, 8, 9, 2$ and 12, in that exact order as in Figure 8, then 64 of the images of the true label 0 were predicted by the model to have the label 0, 12 were predicted to have the label 1 and so on. This means the perfect identification model would generate a confusion matrix with non-zero numbers on the diagonals and zeroes everywhere else.

III. Method

In this section we present the methodology of this project. The relevant code is found in [8].

A. Programming Language, Tools and Machine Learning Algorithms

The programming language used in this project was Python. Python is a versatile programming language, which includes a great variety of additional machine learning tools to choose between. Moreover, we mainly used the tools from Scikit-Learn [9].

The number of different machine learning algorithms that can be used for image recognition, and classification problems in general, is vast. Given the limited time allocated for this project and the prior knowledge of the members of the project group, *random forest* was found to be a suitable machine learning algorithm. This algorithm is common and not very hard to implement, especially with the help of Scikit-Learn, and it also performs well in a variety of classification problems. *Central Neural Networks* was also looked into, but was chosen to be excluded from this project.

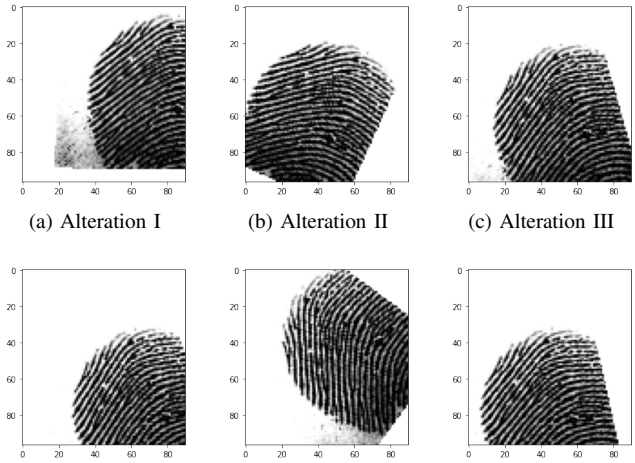
B. Dataset Expansion

As previously described, the dataset had to be expanded such that the machine learning model could be properly trained. The strategy for this expansion that was mainly investigated, was shifts and rotations of the images. That is, from each fingerprint image, a large number of copies were created. These were then rotated by an angle, which is a realization of a Gaussian distribution with zero mean and standard deviation of $30°$. The copies were also shifted horizontally and vertically by numbers of pixels, which are realizations of a Gaussian distribusion with zero mean and standard deviation of 20% of the height and width of the images respectively.

These alterations made each copy differ slightly from its original image, hence increasing the size of the training set. For simplicity, we did not consider other more complex alteration scenarios, such as cuts in the fingerprints or blurry images.

The motivation behind the expansion by rotations and shifts was that these types of variations in the images are close to real-life scenarios. When a fingerprint is scanned, it will most probably not be positioned exactly the same as the original image that the model was trained on. Since Gaussian distribution is the most common distribution found in nature, it was a natural choice of distribution for both the rotations and shifts. The standard deviation for angles and numbers of pixels to shift by were picked by logical reasoning. These can reasonably be common magnitudes of the alterations in real life. Examples of altered images are presented in Figure 5.

Since this dataset expansion method is not studied in the literature to the best of our knowledge, its validity had to be investigated. Most commonly, when a machine learning algorithm is trained, the training set consists of a vast number of different images for each category, and the testing set is similarly constructed. This as opposed to the SOCOFing dataset, which only contains one image per fingerprint.



(a) Alteration I (b) Alteration II (c) Alteration III

(d) Alteration IV (e) Alteration V (f) Alteration VI

Fig. 5. Images of altered fingerprints

As the goal was to synthesize a dataset as similar as possible to the standard kind of training/testing set that includes a large number of different original images for each category, we had to compare the performance of our model when it had been trained on a standard training set to its performance when it had been trained on a synthesized training set. This was done using the MNIST dataset [6].

The training set of the MNIST database consists of 60000 images of handwritten digits. By randomly choosing one image from each category (digit) in the MNIST training set, we could use those 10 images to generate a new training set in the same manner as we did with the SOCOFing dataset. Testing was then performed on the rest of the images in the original MNIST training set.

To evaluate the performance of the trained models using the synthesized dataset, we compare the testing accuracy with that of the trained model using the regular training dataset with different images. The comparison results are plotted in Figure 6, where the standard training data set includes 80% of the whole MNIST dataset.

C. Single-Stage Identification

Because of the limited computing power of the computers used in this project, the whole set of images in the SOCOFing dataset could not be used. Instead, the results from the dataset expansion helped determine the smallest number of altered copies sufficient for training the model.

It was decided that a dataset of 100 original fingerprints from 10 individuals, expanded by 100 altered copies per original image, was sufficient for our purposes. Based on this synthesized dataset, the resulting confusion matrix is plotted in Figure 8.

D. Two-Stage Identification

For the two-stage identification, the same dataset as in the single-stage identification was used. That is, a dataset of 10000 images generated from 100 original fingerprint images from 10 different individuals was used.

In two-stage identification, the identification process is divided into two steps. Firstly, the model classifies a more general category of the fingerprint. Then, the specific identity of the fingerprint.

The classification categories that were investigated in this project were *gender*, *hand label* and *finger label*. That is, the fingerprints were either classified by: the gender of the individual in question, i.e. male or female; the hand that the fingerprint belonged to, i.e. left or right; or the finger that the fingerprint had been scanned from, i.e. thumb, index finger, middle finger, ring finger or little finger. These were categories that were already provided by the dataset, which was useful since we were working with supervised learning.

To get an overview of how well the model could classify fingerprint images into these categories, classifications were first performed on the full, non-extended, original SOCOFing dataset of 6000 images, and the dataset was split into 80% training data and 20% testing data. The results are depicted in Table I and Figures 9, 10 and 11.

The full two-stage model was implemented as described in Section II-F. The results were evaluated using cross-validation mean accuracy and confusion matrices.

IV. Results

A. Dataset Expansion

1) MNIST: In Figure 6, the test accuracy versus the number of altered copies of the training image for the MNIST dataset are depicted. The accuracies are given as a mean of evaluations from five different training sets, where each training set is generated from the same original image. The alterations in the training sets differ slightly, as an effect of the randomness in the generation process.

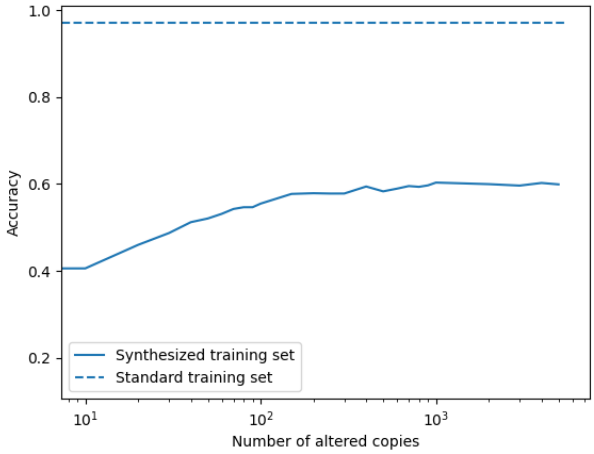


Fig. 6. Mean accuracy of predictions versus number of altered copies of each digit in the training division of the MNIST dataset. Note that the x-axis is in logarithmic scale.

The standard training set is non-extended and consists of 48000 randomly picked images from the MNIST dataset. The accuracy of the model that had been trained on this training set amounted to 0.97 and is seen as the upper, dashed line in Figure 6.

2) SOCOFing: Figure 7 depicts how accuracy varies according to the number of copies for the SOCOFing dataset. The graph was generated using a subset of the full dataset, consisting of fingerprints from right-hand thumbs of 10 different individuals.

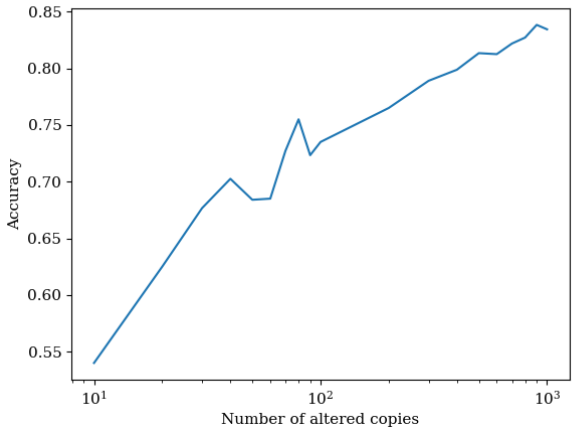


Fig. 7. Mean accuracy of predictions versus number of altered copies of each fingerprint data sample in the SOCOFing dataset. Note that the x-axis is in logarithmic scale.

B. Single-Stage Identification

The result of the single-stage identification is depicted in the confusion matrix of Figure 8. Five-fold cross-validation gave a mean accuracy of 0.27, with a standard deviation of 0.03.

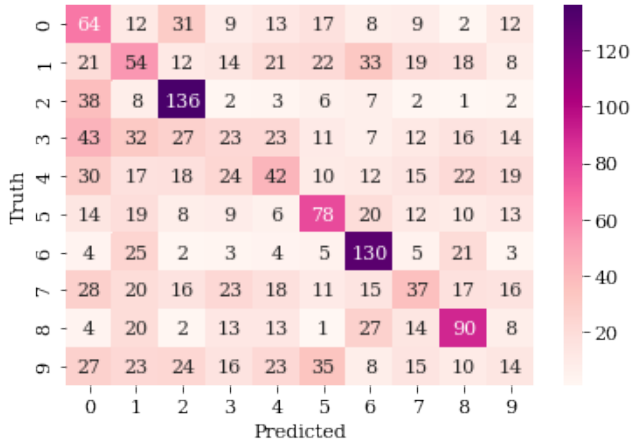


Fig. 8. Confusion matrix for prediction of identity using single-stage identification.

C. Single-Stage Classification

The cross-validation accuracies of the classifications, using the non-extended original SOCOFing dataset, are listed in Table I. Confusion matrices for the classifications are found in Figures 9, 10, and 11.

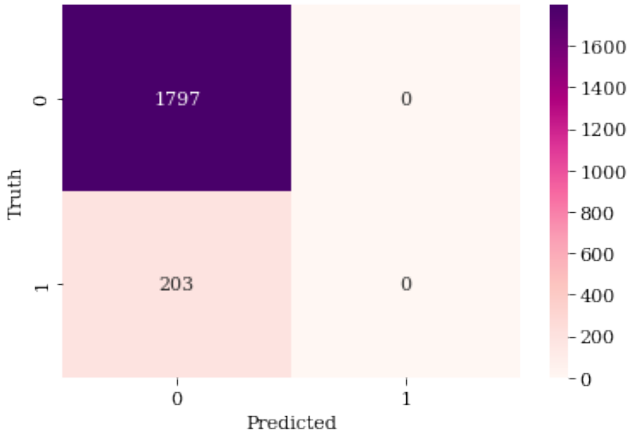


Fig. 9. Confusion matrix for prediction of gender label using non-extended original dataset. 0 corresponds to male and 1 to female.

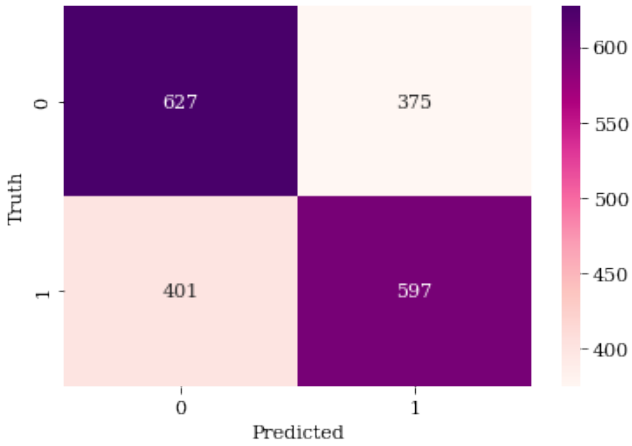


Fig. 10. Confusion matrix for prediction of hand label using non-extended original dataset. 0 corresponds to the left hand and 1 to the right hand.

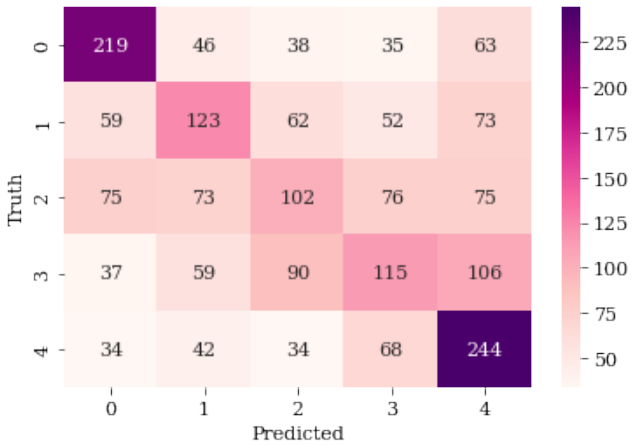


Fig. 11. Confusion matrix for prediction of finger label using non-extended original dataset. 0 corresponds to thumb, 1 to index finger and so forth.

TABLE I
5-FOLD CROSS-VALIDATION ACCURACIES FOR DIFFERENT CHOICES OF SINGLE-STAGE CLASSIFICATIONS.

Category	Mean accuracy	Standard deviation
Hand	0.72	0.01
Gender	0.80	0.00
Finger	0.49	0.01

D. Two-Stage Identification

The confusion matrix for the two-stage identification with classification by hand label is given by Figure 12. The confusion matrix for the two-stage identification by gender label is given by Figure 13 and a confusion matrix for the two-stage identification by finger label is given by Figure 14.

The mean accuracies and standard deviations of 5-fold cross-validations of the two-stage identifications are summarized in Table II.

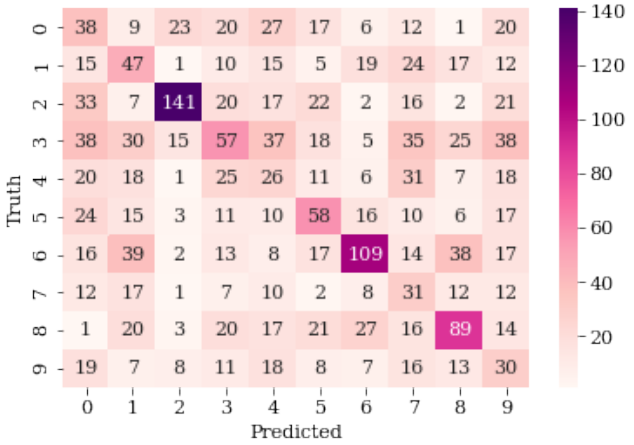


Fig. 12. Confusion matrix for prediction of identity with 2-stage identification, categorizing by hand label in first stage. 0 corresponds to the left hand and 1 to the right hand.

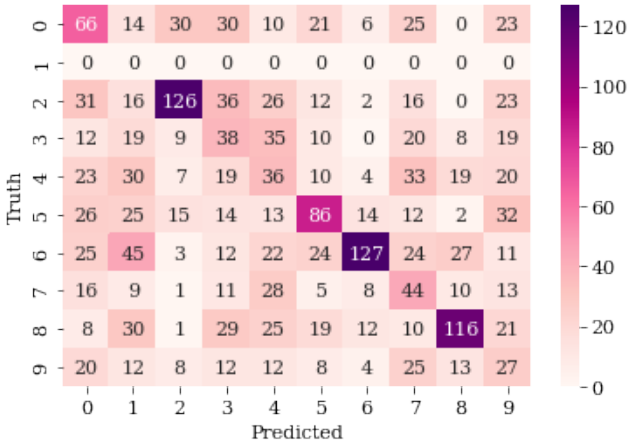


Fig. 13. Confusion matrix for prediction of identity, 2-stage identification with gender label categorization in first stage. 0 corresponds to male and 1 to female.

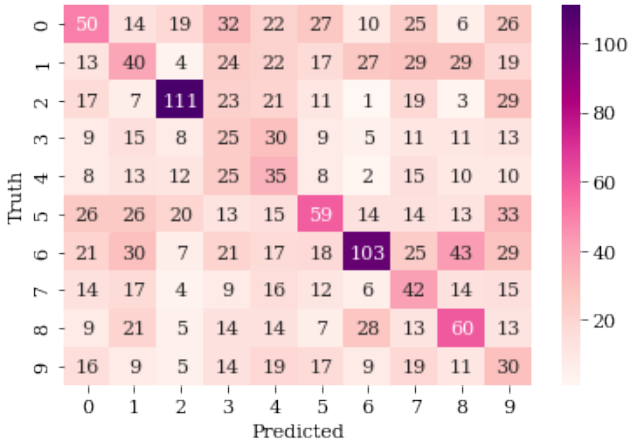


Fig. 14. Confusion matrix for prediction of identity, 2-stage identification with finger label categorization in first stage. 0 corresponds to thumb, 1 to index finger and so on.

TABLE II
5-FOLD CROSS-VALIDATION ACCURACIES FOR DIFFERENT CHOICES OF TWO-STAGE CLASSIFICATIONS.

First stage categorization	Mean accuracy	Standard deviation
Hand	0.31	0.01
Gender	0.34	0.01
Finger	0.27	0.01

V. Discussion

A. Dataset Expansion

Observing Figure 6, we see that the accuracy increases as the number of altered copies of the synthesized training set increases. Even when only one image is used for training, which is depicted in the leftmost part of the graph, the random forest exceeds mere coincidence. The accuracy then increases as the number of training images increases, up to a peak value of approximately 0.6. Compared to the accuracy of 0.97 when the algorithm is trained on a standard training set, this result is notably worse. The increasing accuracy does, however, indicate that the method of rotating and shifting images can improve the performance of an algorithm when the quantity of the training data is limited.

If we compare Figure 6 with Figure 7, we find that the graph in Figure 7 does not reach an as obvious peak value, but rather tends towards the infinite accuracy of 1.0. This is most probably caused by the fact that the greater the number of copies in the training set, the greater the chance that the exact same image being tested on is already in the training set. This is a contrast to the MNIST dataset, in which the testing images had no chance of also being in the training set. As such, the more copies in the training set, the greater the chance of overlaps between the training and testing set, and naturally the accuracy must increase until all image alterations are both in the training and testing set.

Lastly, one can note that the method of expansion is not exhaustive. As was noted in Section III, the choice of rotating and shifting the images came from a wish of extending the

dataset as naturally as possible. There will, however, always be other distortions in new fingerprint scans as well. The skin is flexible and might become slightly distorted compared to the image in the dataset. In addition, there might be dirt or greasy stains present, or somebody might have injured their finger, leaving parts of the fingerprint changed. Thus, a dataset of solely shifted and rotated copies of a few fingerprint images will not reflect all of the variations in a naturally generated set of fingerprints.

B. Single-Stage Identification and Classification

Figure 8 depicts a confusion matrix for single-stage identification. This confusion matrix is the desired diagonal matrix which, together with the accuracy of 0.27, suggests that the identification can very well be successful to some extent.

This can be compared with Figure 9, in which the pattern is not diagonal. The high accuracy of 0.8 seems not to be based on good predictability of the algorithm, but rather from the fact that 80% of the images in the dataset were from male subjects, making it most favorable for the algorithm to simply suppose all fingerprints belonged to males. This explains why the confusion matrix is heavy on the left.

The confusion matrices for the prediction of hand and finger labels are depicted in Figure 10 and Figure 11, respectively, and they both have the desired diagonal pattern. It seems that the algorithm distinguishes the thumb and little finger categories better than it distinguishes index fingers, middle fingers and ring fingers. The majority of the samples are, though, correctly classified. The same goes for the hand label classification.

C. Multi-Stage Identification

The accuracies of the two-stage identifications are slightly improved compared to the single-stage categorization when considering the hand and gender classification versions. For the finger classification version, the accuracy is precisely the same. Considering the small size of the training set, these results should not be taken too seriously. As was seen in Figure 9, the classification by gender did not function properly.

VI. Conclusion

Using rotation and shifts to extend the given SOCOFing dataset shows good performance, giving a higher accuracy of prediction with a larger number of data samples.

Single-stage and two-stage identification perform approximately equally well and more research has to be carried out to conclude on the advantages and disadvantages of the two setups.

VII. Possible Further Research

Possible future research could include the use of unsupervised learning and the use of the supercomputer at the KTH Royal Institute of Technology. The latter would make it simpler to exert a larger share of the SOCOFing database. It would also be profitable to analyze the time complexity of the different setups.

Acknowledgment

The authors would like to thank Linghui Zhou for her great feedback and helpful opinions.

References

[1] A. Kantarci, “Image recognition in 2021: In-depth guide,” Jan 2021. [Online]. Available: https://research.aimultiple.com/image-recognition/

[2] F. Farhadzadeh and F. M. J. Willems, “Identification rate, search and memory complexity tradeoff: Fundamental limits,” *IEEE Transactions on Information Theory*, vol. 62, no. 11, pp. 6173–6188, 2016.

[3] M. T. Vu, T. J. Oechtering, and M. Skoglund, “Hierarchical identification with pre-processing,” *IEEE Transactions on Information Theory*, vol. 66, no. 1, pp. 82–113, 2020.

[4] (2020, Dec) Random forest. [Online]. Available: https://www.ibm.com/cloud/learn/random-forest

[5] Y. Shehu, A. Ruiz-Garcia, V. Palade, and A. E. James. (2018, Jul) Sokoto coventry fingerprint dataset. [Online]. Available: https://www.researchgate.net/publication/326681401_Sokoto_Coventry_Fingerprint_Dataset

[6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. (1998, Nov) Gradient-based learning applied to document recognition. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[7] F. Willems, T. Kalker, J. Goseling, and J. P. Linnartz, “On the capacity of a biometrical identification system,” *Proc. IEEE Int. Symp. Inf. Theory*, no. 1, p. 82–87, Jun 2003.

[8] H. Israelsson and A. Wrife, “Kex2021,” 2021. [Online]. Available: https://github.com/hannaisr/KEX2021.git

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

CONTEXT O
COMPUTATIONAL BRAIN MODELLING & BRAIN-LIKE COMPUTING

POPULAR DESCRIPTION

Thinking About how a Thinker Thinks

The brain has always been curious about what is going on inside itself. We can think abstract thoughts. We can remember what happened years ago. We can even fantasize about the future. Although this thinking-machine has been contemplating itself for thousands of years, we know less about the brain than we do about the world's oceans. However, the development of computers allows us to create simulations that help us understand how our brains function.

Constructing computer programs inspired by the structure and functions of biological brains allows us to study them from completely new perspectives. It could allow us to investigate the brain in ways that might otherwise be impossible or require invasive experiments. Such knowledge would hopefully be useful in order to understand, suppress, and possibly cure neurological diseases. This field of research would therefore have the potential to reduce suffering for many people.

Other than understanding how the brain works, we can also use the knowledge of how the brain thinks and functions to design non-biological intelligence. After all, evolution has already made the biological brain into a very well developed system, so why not take inspiration from it? The field of artificial intelligence tries to create intelligent systems with the ultimate goal of developing machines with at least human-level intelligence. Brain-like computing thereby has the potential to play an important part in the artificial intelligence of the future. It could also have the potential of enhancing humans' bodies and minds by offering a connection that allows the brain to interact directly with computers and prosthetics. In its limit, brain-like computing and computational brain modelling could allow us to finally accomplish one of humanity's oldest dreams: to understand the most complex machine that we know of—our own thinking brain.

SUMMARY OF PROJECT RESULTS

The context of computational brain modelling and brain-like computing originates from studying the structure and functions of the human brain. Computational brain modelling aims to construct computational models that can be used to study different aspects of the biological brain and its behaviour. These aspects can range from high-level structural models used to study cognitive functions, to detailed models accounting for connections between individual neurons. The levels of biological abstraction may also be varied in the models. Brain-like computing builds upon the models developed in computational brain modelling and takes this one step further by designing algorithms that take inspiration from how brains operate. While the problems are generally similar to those that traditional machine learning tries to solve, the brain-like computing algorithms are often very different from deep learning neural networks, currently popular in machine learning. Consequently, they offer many interesting advantages but also pose new emerging challenges.

The research in this context was conducted using existing network models from previous research, implemented from scratch in Java and Python by the project groups. The network architectures and mechanisms used are inherited from existing network models of the brain's memory systems, and the results could be used to investigate the properties of these models further as well as drawing comparisons to biology. The networks used by both project groups are examples of the *Bayesian confidence propagation neural network* (BCPNN). This learning rule is governed by the Bayesian probabilistic framework and inspired by Hebb's rule: "neurons that fire together wire together", meaning that the strength of the connection between two neurons depends on the probability of both being stimulated simultaneously.

The project group O1 implemented a BCPNN learning rule in Python to simulate long term memory. The purpose was to study how the simulated memory was affected by computerized equivalents of varying biological learning environments, in tandem

with varied behaviour of the network's ability to adapt to input. The network's ability to recall memories was studied by observing how many correct recalls were produced either when varying the network's plasticity parameter or the number of repetitions in learning. Both aspects were applied only for selected "years" in a simulation of a lifelong memory.

The results obtained by project group O1 include a recreation of a phenomenon found in the human brain known as the *reminiscence bump,* entailing that memories from adolescence to early adulthood more easily are recalled than ones from other time periods. It was also possible for the project group to observe that the modulation of network plasticity affected the recall of memories in comparison with no modulation. The difference was also more noticeable for early simulated years than for later ones. Furthermore, repetition of selected patterns showed an impact on the ratio of correctly recalled memories over time.

Previous research into long-term memory effects, such as the *reminiscence bump,* has focused on the origins of the phenomena. From a biological perspective, the results obtained by varying the plasticity in a network provide a general idea of how the upregulation of brain plasticity due to stress or other emotional factors can impact the brain's learning and memory characteristics. The plasticity directly impacts the strength of encoding of a specific memory pattern, thus making it a long lasting memory.

There are many questions left to be answered about the cognitive characteristics of biological neural systems. Regarding the model of long term memory, one possible expansion of the project would be to take into account that certain aspects of memory are more strongly encoded than others. For example, the sight aspect is generally stronger than the other sensory aspects of a memory. Translating this into a computer model could mean varying the network plasticity for different parts of the memory pattern. As the goal of computational brain modelling is to create biologically correct models of certain aspects of the brain, this development would contribute to that goal in regards to accurate modelling of memory.

The project group O2 investigated the sequence learning capacity (meaning how much sequential information the network can learn) of a recurrent *Bayesian confidence propagation neural network* with two time-traces of synaptic activity (one that changes faster and one that changes slower). The network consisted of network units organized into groups called *hypercolumns* within which there was a certain competitive mechanism for determining which units are active (the so-called winner-takes-all mechanism where only the unit in each hypercolumn with the highest synaptic activity is activated). This network architecture has previously been used to model both short- and long-term memory effects in the human cerebral cortex. In particular, the network's ability to learn more and longer sequences with different amounts of overlap depending on the size and structure of the network was investigated. When training and testing the network, each symbol in the sequences is mapped to a unique pattern in the network.

Group O2's experiments have shown that the sequential information storage capacity of the network is strongly positively connected to hyper column size. Specifically, the network's sequence disambiguation quickly becomes less reliable if the number of units per hypercolumn decreases below the number of unique symbols used in the set of sequences used. This forces the patterns corresponding to different symbols to have some overlap in the form of shared states of some hypercolumns (this is referred to as spatial overlap). Increasing the number of hypercolumns also helps with disambiguation and makes it easier for the network to handle spatial overlap.

Biological networks are often very large, and practical applications of brain-like computing models are likely to require correct recall and disambiguation of large amounts of data. Previous research on the network model used in project O2 has focused primarily on the qualitative behaviour of smaller models in very specific cases. The results of this project make a contribution towards the goal of understanding how the behaviour of the network changes when it is scaled up, and how different types of data affect the network's behaviour. In this, the work also contributes by investigating how more than two overlapping sequences can be learned, which is something that has proven difficult in several previous studies.

Looking at the model of memory in topic O2, research needs to be conducted on how such networks can be used on a much larger scale in order to learn to disambiguate and store enough sequential information to reflect human memory. Research also needs to be conducted on how the networks can be connected to other parts of a cognitive system, which is far beyond the scope of project O2. Concretely, it might be interesting to investigate how several instances of the model used in project

O2 could be connected with other—possibly brain inspired—models by synapses to specific units in these instances. In such a setup the brain-like models could potentially remember sequences and work as short-term memory together with other networks without that ability. As previously mentioned, another interesting direction for research could be to investigate if it is possible to find a more firmly biologically based model (for example in terms of hyperparameter choices) that also scales well.

IMPACT ON SOCIETY AND ENVIRONMENT

There are multiple aspects of brain-like computing and computational brain modelling that are important to consider from a wider perspective, taking into account environmental, societal and ethical points of view. Although brain-like computing and computational brain modelling does not directly affect the natural environment, they pose a risk for indirect impact through energy consumption due to large computations that require large amounts of energy to run. If the energy is not produced in a sustainable way this could have a negative environmental impact. Since the brain performs computations very efficiently compared to current artificial computer hardware, it might be possible in the future to design neuromorphic hardware which offers very low-energy computational solutions.

From a societal point of view, bridging the gap between the empirical findings and theory about how the brain works has a number of advantages. By getting a greater understanding of how the human brain works, it is not only possible to further our knowledge of how we function as humans on a more philosophical level, but it is also possible to find what causes many neurological diseases. This is because many of these diseases manifest themselves in specific cognitive functions, such as memory. With computational neuroscience one might be able to identify biomarkers that give early indications of these diseases. Through this acquired knowledge there is potential to develop treatments and preventative measures in order to slow down the course of events of, or even prevent these diseases from emerging altogether.

Another possible impact would be changes in how neurological research will be conducted. While it is important to have a basis in the underlying biological systems, a computer-simulated model of a particular part of a brain, shown to accurately represent selected neural phenomena, could be used as a complement to experiments on animals or humans. Where we today do tests on animal brains or with living humans, we could partly resort to computational models and thus achieve more ethically sound, efficient, and large-scale experiments.

With brain-like computing, robots with more human-like qualities could potentially be designed. This could aid in making those robots more trustworthy to humans so that they could more easily be integrated into human society and help people in their daily lives. A problem with most human-like robots today is that they often fall into the uncanny valley of almost looking, behaving, and sounding like humans. However, in at least one of these aspects they are still perceived as slightly too inhuman, making them very unsettling. Robots constructed to think like humans and exhibit some of the same flaws as us have the potential to be less unsettling, making it easier for them to be accepted and used in society.

The development of computational systems based on an increased understanding of the human brain can pose social problems even if they are not near or above human levels of intelligence. For example, if social media companies or government agencies have access to much better models of human behaviour than today it would give them more predictive and manipulative power over users and citizens. To combat this, more regulations (and discussions in society) regarding surveillance, privacy, personal data, and the organizations possessing powerful artificial intelligence technologies are likely to be urgently needed, among other things.

The development of increasingly intelligent systems also poses serious risks. If we were to develop artificial intelligence systems, which greatly surpass our human capabilities in all relevant aspects—something known as superintelligence—both the potential benefits and risks would be immense. One fundamental problem that needs to be solved to avoid dangerous outcomes is known as the *alignment problem*, which is the problem of aligning the goals of the superintelligence with those of humanity. While defining “the goals of humanity” is a very hard problem in itself, a superintelligence constructed with a basis in a brain-like architecture could potentially make it easier to align the goals since its workings would be somewhat similar to ours. The dynamic of introducing superintelligence to our societies would likely affect the structure of society in unpredictable ways, depending on the nature of the superintelligence. We are most likely currently far from approaching the

actualisation of these scenarios, but due to their potentially existential nature anyone working in this direction should keep them in mind. More research needs to be conducted in the field of AI-safety to figure out good solutions to these problems before they actually arise.

A positive effect of a greater understanding of the human brain and the development of artificial neural networks similar to biological ones could be the possibility of integrating our human brains with computers. Such a development has the potential to be more satisfactory to many people, rather than a potential alternative: simply being surpassed (for example at their jobs) by an artificial intelligence that becomes more capable than a human being. This could possibly open the door to expanding the human experience, and prolonged life by protecting the content of the brain from biological decay. However, there are ethical concerns to consider regarding this, such as safety aspects and the risk of people feeling pressured to artificially augment themselves to stay competitive in the work environment or society in general.

One common criticism of other artificial intelligence systems such as deep-learning networks is that they are hard to understand. If they make a decision, researchers usually do not know why it made that decision. While they have empirically been shown to generalize well and adapt to “new” input, examples of inputs that produce unexpected outputs, effectively fooling the system, can be found relatively easily. Some brain-like computing approaches, especially those built to exploit the Bayesian apparatus, offer artificial intelligence solutions that are fundamentally easier to probe and understand. They also allow for modular solutions where different models can be combined in a way, which is uncommon for other artificial intelligence approaches. This further facilitates an understanding and interpretation of the system.

The Impact of Selective Plasticity Modulation on Simulated Long Term Memory

Alicia Palmér and Silvia Barrett

***Abstract*—Understanding the brain and its functions is a challenging undertaking. To facilitate this work, brain-inspired technology may be used to examine cognitive phenomena to a certain extent, by replacing real biological brains with simulations. The aim of this project was to provide insights into how different kinds of plasticity modulation affected long-term memory recall through the use of a computational model. A neural network was constructed based on the existing Bayesian Confidence Propagation Neural Network (BCPNN) model and trained with binary patterns representing memories acquired over a lifetime. By varying network plasticity parameters for selected patterns and performing recall of "aging" memories, greater effects were observed in recall statistics for modulation early in the lifetime in comparison with modulation of later ages. From the experiments conducted in this study it was possible to conclude that selective modulation of learning affected the long-term recall of all memories in the simulation.**

***Sammanfattning*—Att förstå hjärnan och alla dess funktioner är en stor utmaning. För att underlätta detta arbete kan hjärninspirerad teknologi i viss utsträckning användas för att studera kognitiva fenomen, genom att ersätta biologiska hjärnor med simuleringar. Syftet med denna studie var att ge en insikt i hur olika typer av modulering av synaptisk plasticitet påverkade ett simulerat biologiskt långtidsminne genom användning av en datoriserad modell. Ett neuralt nätverk implementerat med en inlärningsregel av typen Bayesian Confidence Propagation Neural Network (BCPNN) konstruerades och användes för att träna och återkalla binära mönster, representerande minnen förvärvade under en livstid. Nätverkets synaptiska plasticitet varierades under träning av utvalda mönster och därefter utfördes återkallning av "åldrade" minnen. Testerna påvisade effekt på nätverkets förmåga att korrekt återkalla lagrade minnen. Det visade sig även att modulering utförd på tidiga simulerade åldrar jämfört med modulering av senare åldrar under livstiden hade större påverkan på långtidsminnet. Från resultaten var det möjligt att konstatera att selektiv plasticitetsmodulering under inlärning påverkade nätverkets förmåga att korrekt återkalla samtliga binära mönster i simuleringen.**

***Index Terms*—BCPNN, Long-term memory, Computational brain modeling, Reminiscence bump, Plasticity modulation.**

Supervisor: Pawel Herman

TRITA number: TRITA-EECS-EX-2021:191

I. Introduction

Memory is an essential part of everyday life. It follows us throughout all chapters of our lives and allows us to capture the experiences that we encounter in the world. It is also plays a key part in many of the brain's functions. This makes memory an important subject for research, since evaluating the function of our brain and the way that it creates and stores memories could in the future be a key to revealing underlying causes of memory-affecting brain disorders, such as Alzheimer's disease [1].

Today, the research into the brain's memory function is not exclusively conducted with experimental approaches on living creatures, commonly exploited in psychology and cognitive neuroscience. One can adopt a computational approach to model and simulate specific aspects of memory artificially, using neural network structures. The models themselves may also be implemented with different levels of neurobiological abstraction depending on what brain characteristics one wishes to study. Therefore, by taking advantage of technology, one is able to conduct large-scale simulations with full control over network parameters, which facilitates reproducible and systematic studies.

Since the focus of this study was on long-term memory and its properties, a computational model with long time constants appropriate for this purpose was chosen. In previous studies, computational models have been constructed to reflect long-term memory phenomena, such as the decay, inhibition and overload of memories [2] in the same manner as it appears in biological memory. One such model, who's behaviour and learning is described in terms of first-order differential equations, is the BCPNN model. This network model operates in resemblance with the brain on a neuronal level, unlike other models such as the Tracelink and the Memory-chain models [3] [4]. Because of the BCPNN model's ability to relatively realistically reflect a brain's long-term memory function, this model was chosen for this study. "... [BCPNN] can be seen both as an abstract learning model and a working hypothesis regarding cortical associative memory function" [5].

A. Project background

This study aims to contribute to the exploration of long-term memory phenomena with focus on the effect of selective plasticity modulation on the memory performance. Different test scenarios were designed using the BCPNN memory model in order to determine how varying plasticity parameters in the network affected the retrieval of long-term memories. A well-known feature of human long-term memory is the cognitive effect known as "the reminiscence bump", showing that the ability to retrieve memories is dependent on the age at which memories are created, with memories from the ages 10-30 years being the easiest to recall [6].

Multiple computational studies of long-term memory phenomena have been conducted using the BCPNN learning rule. Several of these accounted for the reminiscence bump and reported how it depends on different parameters in the BCPNN model itself [7]. However, it has not been investigated how varying plasticity parameters for certain ages affects long-term memory recall. The results of this study may therefore provide a seed for upcoming research in the area of long-term memory.

B. Research question

This project aims to answer the question as to how selective BCPNN plasticity modulation affects long-term memory recall characteristics and effects such as the reminiscence bump. The term plasticity in this context refers to the flexibility of the network connections in the BCPNN model. This question can be studied with respect to different aspects of plasticity modulation, for example a repetitive stimulation of selected memory patterns or upregulation of their learning rates. Both of these aspects were considered in this study as they operate on plasticity in manners that are not directly comparable within the network learning rule.

C. Scope

The study was limited to only consider long-term memory, simulating memories using patterns stored in a neural network implementing specifically a BCPNN learning rule. The network was not implemented to simulate interactions between different parts of the brain, but to represent abstract populations of neurons in the cortex.

Due to the requirement of large amounts of data and immense computing power to simulate long-term storage of episodic memories created in a lifetime, the study was limited to representing several memories in one pattern.

The network model used is rather abstract and it relies on rate based units rather than more biologically plausible spiking neurons. In a firing-rate based model, the average neuronal activity is measured over a short period of time and repeated for several runs. Using an abstract model relieves the need for biological details that at the behavioural level are hypothesised to be less relevant. There is also little biological evidence that would help constrain more complex models in the context of the simulations. The focus is instead directed on mechanisms and parameters that are assumed to be of key importance to the simulated phenomena. This way it is also possible to circumvent a challenge of extensive computational load associated with detailed simulations of a spiking neural network model.

D. Objectives

In order to answer the formulated research question the following tasks were carried out:

- Implementation of a long-term memory model capable of storing and recalling patterns representing a lifetime
- Designing a training algorithm capable of applying selective plasticity modulation
- Designing experiment scenarios to test memory recall ability
- Evaluation of outcome for the different test scenarios

II. Background

The methods used in this study build upon previous work on episodic BCPNN memory models. These studies present the formulas that control the dynamics of the learning process [8] [9], as well as network parameter values that allow the models to realistically reproduce the characteristics of synaptic connections in the brain [10]. The models themselves are based on the use of neural networks and manifest the properties of human long-term memory.

Related work includes one study that implemented BCPNN models to simulate human long-term memory, in order to evaluate how the phenomenon of reminiscence bump changed with varying network parameters [11]. This study did however not include selective plasticity modulations, as further considered in this project.

This sections introduces the basics of the human brain and long-term memory, and further describes their relevance in terms of the implementation of a biologically plausible BCPNN learning rule.

A. The human brain and memory

The human brain can very briefly be described as a composition of individual signaling nerve cell units known as neurons. These can obtain different states, being either active or inactive depending on incoming stimuli [12]. Through extensions of the cell body known as axons, neurons form connections with other neurons. The axon, sometimes surrounded by an insulating layer called myelin, will forward electrical impulses from the cell body and the resulting message will travel through the axon to eventually chemically stimulate the target cell through a chemical terminal known as a synapse [12]. The chemical stimulation will then either allow the electrical impulse to be forwarded into the next cell or be prohibited. This process will result in communication between the two neurons. An example of two connected neurons is depicted in Fig. 3 A).

Unlike in an electrical circuit, the connections of the “wires” are not constant. The synaptic connections are dynamic and vary both in time and depending on neural activity and other factors. The synaptic transmissions can due to this be either strengthened or weakened [12]. One may refer to this property as synaptic plasticity. With age, the number of synaptic connections in the brain decreases. This can result in a deteriorated ability to make associations and could for example be being able to pair a name with a face or to remember details of previous events [12].

The anatomy of the brain is complex and includes different parts as well as different layers. The cerebral cortex, referred to simply as cortex from this point onward, is the outermost layer of the brain and one of the most remarkable elements of the human brain [13]. It contains up to 20 billion neurons and possesses functions of memory, learning, thinking and consciousness [13].

1) Hebb’s rule

Neuropsychologist Donald O. Hebb is famous for his studies conducted during the 20th century on how behaviour and psychology relates to the neuroscience of the brain’s nervous system. Hebb developed the neuroscientific theory referred to as “The Hebbian rule” [14] [15]. The theory illustrates that a neuron is more likely to become activated, or ”fire”, if an interconnected neuron fires, granted that this mutual form of firing has taken place multiple times previously. This was formulated by Hebb himself as “When an axon of cell A is near enough to excite a cell B and repeatedly or persistently

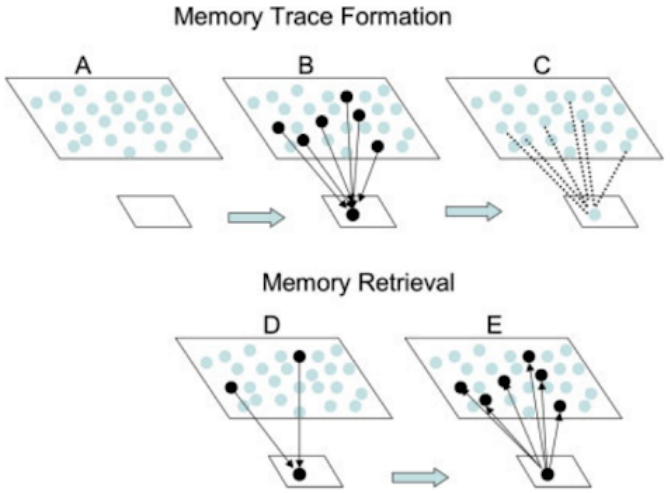


Fig. 1. Memory trace formation: (A) The top layer represents neocortical activity and the bottom layer the hippocampus. (B) A pattern activates, which is projected on the hippocampus. (C) The memory is stored in terms of strengthening the synaptic connections. Memory retrieval: (D) A subset of the pattern activates the hippocampus. (E) The hippocampus projects back and reactivates the whole pattern [18].

takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased" [16]. This theory is often summarized as "What fires together, wires together" [16].

2) Biological memory

The process occurring in the brain in which new information is obtained by the nervous system is referred to as learning, while memory refers to the handling of previously learned material [12]. There are different ways to categorize memory; one example is by the content of what is remembered. When observing how long different memories are stored in the brain, three temporal categories are usually considered [12]. The *immediate memory* and the *working memory* are both forms of short term memory, where information is kept temporarily for time periods ranging from fractions of a second to a couple of minutes. *Long-term memory* is the last category in which memories are stored in a more permanent way, ranging from days to several years [12].

How and where memories are stored in the brain is indeed dependent on both their content and temporal affiliation [12], but in general it is established that memories are stored through the process of strengthening the synaptic connections between neurons [17]. A much simplified explanation of the process to store and recall a memory is illustrated in Fig. 1, in terms of neurons in a layer of the cerebral cortex, titled neocortex, receiving stimulation from another part of the brain known as the hippocampus.

3) Reminiscence bump

The reminiscence bump is a cognitive phenomenon that has been a subject for autobiographical memory studies for many years [7]. It describes the tendency of elderly people having the ability to recall memories from later childhood until early

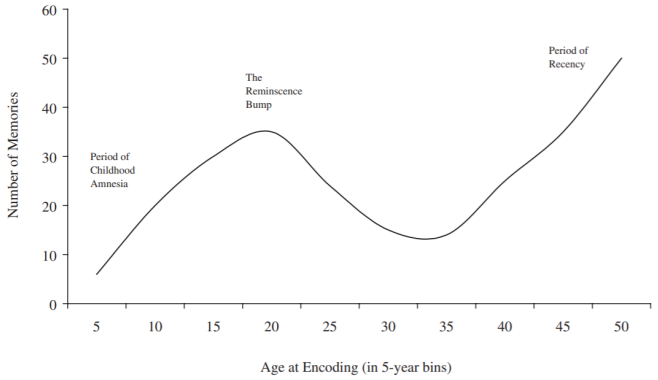


Fig. 2. The reminiscence bump withrecency effect [19]

adulthood better than from other periods of their lives. The actual age span where the bump occurs does vary between studies, but the general age span is usually said to be between ten and 30 years of age [7]. The reminiscence bump is depicted in Fig. 2, which also shows a "period of recency" appearing due to the fact that recent memories generally are easy to recall.

It has not yet been established exactly why this phenomenon occurs, although some hypotheses exist. These include aspects such as the amount of events occurring and their emotional weight during this period of time. A lot of important events taking place during those years are crucial to the formation of self identity [6], making them more emotionally significant than others and therefore perhaps easier to recall in later years. One hypothesis implies that the memories of events during the specified age span are more strongly encoded simply because they are followed by an in comparison tranquil period, giving the brain time to processes these previous events [6]. Another hypothesis further suggests that the first memory of what becomes a recurring event is often going to be the best remembered one, implying that the reminiscence bump could be a result of multiple first-time events during a limited time period [20].

B. Bayesian Confidence Propagation Neural Network

Computational simulations of the process of learning and recalling long-term memories are in this study conducted with the use of a BCPNN model, which is a form of recurrent associative memory network. Recurrent, in this context, refers to the dynamic characteristics of the network, allowing for temporal state representations [21]. The associative aspect refers to associative memory, meaning that the memory is content-addressable [21]. For this reason, it is possible to recall a learned pattern through only partial activation. The BCPNN memory model uses Hebbian-like learning to acquire memory [10].

This section describes the background of how the BCPNN model is constructed, starting from the basics of neural networks. The specific implementation used in this particular study is described in section III.

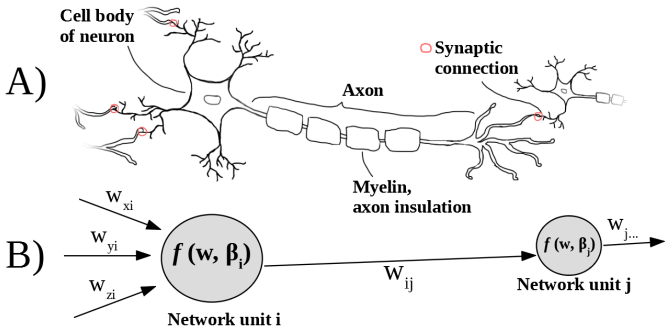


Fig. 3. Analogy between neuron connections (A) and units in a neural network (B)

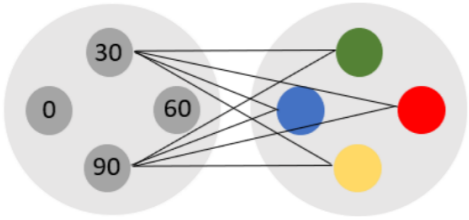


Fig. 4. One module (left) describing the orientation of an object in degrees, and one (right) describing its color. The units represent the possible colors and angles [11].

1) Neural networks and their resemblance to biology

Neural networks have many uses, one of which is to recognize patterns [22]. A network consist of a collection of individual units with intermediate connections. The strength of the connections are controlled by network parameters referred to as weights, and their values depend on the units' relationships to other units in the network. The units can be considered as basic computational nodes in the network, and are on a very abstract level related to neurons in the brain. The analogy between a unit and a neuron is illustrated in Fig. 3.

2) Probabilistic learning

For many types of neural networks, their learning is determined by the actual values of the units. It is however possible to implement learning rules based on probabilities instead by including elements of probability theory, oftentimes Bayes theorem of conditional probability. This theorem describes how the probability of an outcome is based on the probability of related events [9] [23]. BCPNN is a form of modular attractor memory neural network [24]. It applies a Bayesian-Hebbian learning rule, meaning that the learning rule is incorporating probabilities derived from Bayes theorem [10] and follows the Hebbian principle (see section II-A1) in the way that probabilities play a part in activating units in the network [24].

3) Network structure

One unique property of the BCPNN learning rule is its network structure consisting of units divided into subgroups which are called modules. A unit in the network would in the brain correspond to a small group of neurons in the cortex. Multiple units are then further arranged into one module, and multiple modules make up one complete network. This type of network configuration is in line with contemporary theories regarding the cortical function of the brain, suggesting that the cortex consists of multiple subgroups of neurons, behaving like "sub-networks" [10].

One module can be seen to represent a certain aspect of a memory, for example the position or color of a remembered object. Each unit then represents a possible value of the aspect, for example the colors red, green or blue in the corresponding module representing the object's color. An illustration of two modules with their corresponding units can be seen in Fig. 4.

When implementing the BCPNN learning rule, a property known as lateral inhibition will apply to the units. This property ensures that only the strongest unit in each module will determine the value of the corresponding aspect of memory, meaning for example that the color of an object can be either blue or green, not both or more colors at the same time.

4) Synaptic trace variables

Another feature special for the implementation of BCPNN is the use of synaptic trace variables. These correspond to different biochemical processes occurring in the synapses of the neurons [10], behaving in such a way that cell activation is not instant. These trace variables can represent both fast synaptic changes in plasticity to allow for lag between firing neurons, as well as slow changes in memory storage to imitate the decay of synaptic connections in the brain [10].

5) Storing memory patterns

When presented with an input, each unit in the network is forced to become either active or inactive in a way that matches the input pattern's units. This forced activation is referred to as "clamping". Subsequently, the weights and biases get dynamically updated. This means that the particular pattern later can be recalled by clamping only a few of the corresponding active units in the network. Through the recurring connectivity, the remaining units of the pattern should then get activated.

The storage capacity of a network trained with a BCPNN learning rule is limited [25] as for any other form of artificial neural network model of memory, and it is determined by the number of units. This means that the networks only can remember a limited number of patterns correctly. The BCPNN learning rule may however be derived so that it holds palimpsest properties, entailing that the network gradually forgets older memories as new ones are stored and time progresses [24]. This represents the brain's memory function well since people in general tend to remember old memories less clearly than ones from recent events. The palimpsest property is considered one of the major advantages of using a BCPNN learning rule versus other non-biologically inspired artificial neural networks, as those are victims of catastrophic forgetting. This means that they forget all memories once their storage capacity is overridden [24].

6) BCPNN in comparison to other models

Other long-term memory models exist apart from the BCPNN model, also capable of simulating storage and recall of memories. A few examples are the Autobiographical Memory-Adaptive Resonance Theory (AM-ART) network [2], the Memory Chain Model [3], and the Tracelink model [4]. These models do in different ways take into account aspects of memory that the BCPNN-model does not, for example the interaction between different parts of the brain, or the overloading of memories from working memory to long-term memory. However, the BCPNN model is praised for its similarity to neuronal circuits and their behavior, making it one of the most successful ways to simulate biological memory today [11].

III. Method

A. Memory representations

The memories that the network was trained to remember consisted of binary patterns. They were created to have the same number of units as the network. One example of how a pattern was constructed and its categorization of units into modules can be seen in Fig. 5. Each module in a pattern was chosen to have only one active unit. The patterns were generated randomly according to this criteria while also avoiding identical patterns. The average lifetime in Sweden is just over 80 years [26], but due to computational restraints the network was chosen to be trained with 40 patterns, meaning that one memory pattern represents two years of episodic memories in a lifetime. To be able to accommodate all 40 patterns without exceeding storage capacity, Eq. 1 [25] was consulted and a network size of five modules, each consisting of ten units, was used.

$$\text{Max patterns} = \frac{0.79(N^{1.48} - N)}{log_2(N)}. \tag{1}$$

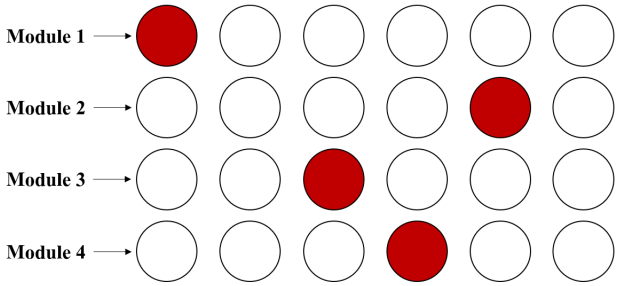


Fig. 5. An example of a memory pattern consisting of four modules with six units per module. The red units are active, corresponding to a one in the binary pattern, and the white units are inactive.

B. BCPNN learning rule

The BCPNN learning rule used in this project is described by Eq. 1-10, all governing the dynamics of unit behaviour in the network,

$$\frac{ds_j}{dt} = \frac{g_w(\beta_j + \sum_i w_{ij}o_i) - a_j + g_I I_j + \sigma - s_j}{\tau_m}, \tag{2}$$

$$o_j = \begin{cases} 1\ , & s_j > s_k, \quad s_k \in M \text{ and } s_k \neq s_j \\ 0\ , & \text{otherwise} \end{cases}. \tag{3}$$

Eq. 2 governed the change in unit support, which can be described as the background activity that controls the activation o, being the output of the network [adapted from [10], Eq. (1)]. This activation was decided by a "winner-take-all" rule, stating that the unit with the highest background activity compared to the background activity of all other units in the same module was declared the "winner". This unit was then assigned a value of one, while all other units were set to zero as described in Eq. 3. To avoid unjustified activation, a condition for the winning unit to be sufficiently greater than all other units in its module was added by including a threshold of 10^{-3} for the difference in support values. If this condition was not met, the whole module was set to zero.

The differential equations describing the behaviour of the synaptic trace variables are descirbed by Eq. 4- 7.

$$\frac{da_j}{dt} = \frac{g_a o_j - a_j}{\tau_a}, \tag{4}$$

$$\frac{dz_j}{dt} = \frac{o_j - z_j}{\tau_z}, \tag{5}$$

$$\frac{dp_j}{dt} = \kappa\alpha(z_j - p_j), \tag{6}$$

$$\frac{dp_{ij}}{dt} = \kappa\alpha(z_j z_i - p_{ij}). \tag{7}$$

The rate of change for the adaptation values was described by Eq. 4 and for the z-traces in Eq. 5. In Eq. 6 and 7 governing both p-traces, the parameter κ was used both as a gate for learning and as a changing parameter in order to answer part of the research question posed. This means that it both acted as a "gate", deciding when training is conducted as well as a contribution of gain in learning when required for appropriate tests. The parameter α described synaptic plasticity decay over time and was determined by Eq. 10 [11]. The weights and biases were governed by logarithmic functions of the p-traces as shown in Eq. 8 and 9. To avoid taking the logarithm of zero, a small constant epsilon was used instead, as described in Eq. 11 [[10], eq. (3)-(8)],

$$w_{ij} = \log_\varepsilon(\frac{p_{ij}}{p_i p_j}), \tag{8}$$

$$\beta_j = g_\beta \log_\varepsilon(p_j), \tag{9}$$

$$\alpha = \alpha_0 e^{-\frac{t}{\tau_p}} + \alpha_{cst}, \tag{10}$$

$$\log_\varepsilon(C) = \log(max(C, \varepsilon)). \tag{11}$$

Tab. I presents the size of the individual networks used along with number of simulations and number of modules inactivated when performing recall. In Tab. II, the values of all network parameters can be seen for the base-case to which the different tests was to be compared. In previous work [10], values similar to these were motivated to be biologically

TABLE I
MEMORY PATTERN PARAMETERS

Name	Value
Modules, M	5
Units, U	10
Network size, N	$M \times U = 50$ units
Number of patterns	40
Number of inactivated modules per pattern (during recall)	2
Number of networks	80

TABLE II
NETWORK PARAMETERS

Name	Description	Value
g_w	gain in weights	1
g_a	adaptation gain	2
g_b	bias gain	1
g_I	input scaling factor	40
τ_P	p-trace time constant	4 $[s]$
τ_M	support time constant	0.01 $[s]$
τ_A	adaptation time constant	0.2 $[s]$
τ_Z	z-trace time constant	0.01 $[s]$
Eulerstep	time per euler step	0.001 $[s]$
μ	mean of noise	0
σ	standard deviation of noise	0.5
κ	gain in learning	1
ε	lower limit for log	10^{-5}
α_0	initial plasticity	3 $[s^{-1}]$
α_C	constant plasticity	0.02 $[s^{-1}]$
Training time	duration each pattern is trained	0.4 $[s]$
Break time	break between training patterns	0.1 $[s]$
Recall time	duration each pattern is clamped during recall	0.02 $[s]$
Convergence time	break between recalling patterns	0.08 $[s]$

Source: adapted from [10]

plausible, although some values were adapted in this project in order to fit the specific implementation. The values of the network variables are defined in Tab. III together with their initial values and dimensions. In a similar fashion, the initial values were adopted from [10].

C. Simulation protocol

As presented in Tab. I, multiple networks were used when testing and the result of one test was calculated by averaging the sum of individual results over all networks. Each network was trained with all of the 40 memory patterns, one at a time. Each pattern was clamped to the network and the weights and biases were allowed to evolve according to the learning rule during the training time, to then settle during break time. Both times can be found in Tab. II. This process is depicted as a timeline presented in Fig. 6, and was repeated for all patterns.

TABLE III
NETWORK VARIABLES

Name	Description	Initial value	Dim.
s_j	Support value of unit j	$\log \frac{1}{U}$	$1 \times N$
o_j	Output value of unit j	$\frac{1}{U}$	$1 \times N$
I_j	Input value of unit j	$\in$ [0,1]	$1 \times N$
p_j	Unit activity trace of unit j	$\frac{1}{U}$	$1 \times N$
p_{ij}	Co-activity trace of connection between unit i and j	$\frac{1}{(U)^2}$	$N \times N$
β_j	Bias of unit j	$g_w \log \frac{1}{U}$	$1 \times N$
w_{ij}	Weight of the connection between unit i and j	0	$N \times N$

Source: adapted from [10]

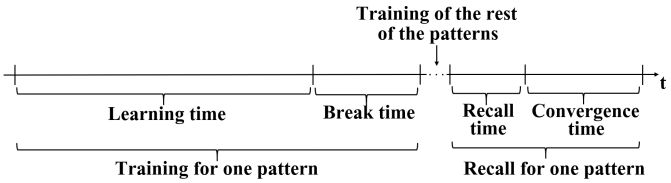


Fig. 6. Timeline describing training and recall protocol. Recall of the rest of the patterns continues beyond the graph.

After one network had been trained with all patterns, a test of the network's memory was conducted by clamping incomplete versions of the trained patterns (see section II-B5). Incomplete version of all trained patterns were created by setting all units to zero in two randomly selected modules in each pattern. These incomplete patterns were then presented in the same order as the original ones were trained, by clamping each one a short period of recall time. After the network had converged during convergence time to a stable state, the unit activation state constituting the output of the network, was recorded. This recall process is depicted for one pattern in Fig. 6. Each output was then compared to all original versions of the trained patterns respectively in terms of Hamming distance. This distance revealed whether the output had converged toward the desired trained pattern or not. To calculate the Hamming distance, each module was compared in the output pattern to the trained one. Dissimilar modules added one to the Hamming distance count and identical modules added zero as exemplified in Fig. 7. This meant that the maximum Hamming distance would be equal to the number of modules in the network, in this case five if all modules differed. If the Hamming distance of one pattern was lower than or equal to the distance of any of the other trained patterns, the recall was counted as successful. This allowed the recalled pattern to contain errors and still be counted as successful, as long as the pattern as a whole was most similar to the originally trained one. The opposite case would mean that the network converged toward the wrong pattern, and the recall would therefore not be considered successful.

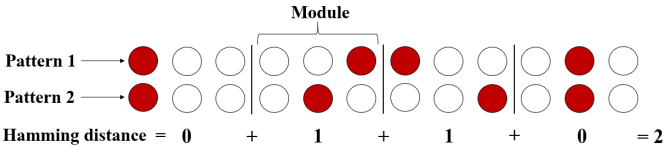


Fig. 7. Example of the Hamming distance between two patterns with four modules of three units. In this case the Hamming distance is two.

The information about which patterns were correctly re-

called was collected, and the order of stored patterns was then randomly shuffled before repeating the same recall protocol with the next network. The shuffling of patterns negated the effects of any particularly "difficult" patterns that would otherwise skew the results. One way that a pattern could be "difficult" is if it shared many active units with other patterns, making them very similar.

The collated data of correctly recalled patterns was then used to calculate the recall ratio for each simulated age. This was done by calculating the average amount of correct recalls (as defined previously in this section) for each respective year over all networks as in Eq. 12,

$$\text{Recall ratio} = \frac{\text{Number of successful recalls}}{\text{Number of network simulations}}. \quad (12)$$

D. Testing paradigm

Testing of how the two types of selective plasticity modulation affected the network's recall characteristics was done in two different types of experiments for each modulation method. The first type was conducted by examining how selective plasticity modulation on different subsets of patterns affected the characteristics of recall. This was performed by training all patterns excluding a subset of three consecutive patterns normally, and then applying plasticity modulations exclusively when training this specific subset. A visualization of this process can be seen in Fig. 8. Different simulations were run where the placement of the subset in the simulated lifetime was changed, but with the use of the same value of κ for all placements. In total, a number of four different subsets of patterns were used corresponding to the ages six to ten, 32 to 36, 50 to 54, and 70 to 74. A recall was then performed as explained in section III-C, in the same order that all patterns were trained. The locations for the modulation subsets were chosen in order to cover different time periods in the simulated lifetime. The two earliest subsets were placed so that one covered the time before the peak of the bump and the other one to cover the time at the negative slope of the bump. The third subset was chosen to cover the period of time around the middle of the simulated lifetime. The last subset then covered the final years, including part of the recency period.

The second experiment type was to test the effects of varying long-term plasticity modulation in tandem with selective modulation. This consisted of the same test as the one previously describes, but with the addition of varying the network parameter α_0 for all patterns including the ones of the subset. Two different values for α_0 were used, 3.4 and 2.6, one higher and one lower than the base-case value of 3 presented in Tab. II.

The two types of experiments were done correspondingly for an increased number of repetitions in learning as well. This meant that the patterns in the subset were clamped to the network several times and therefore trained multiple times before moving on to train the next pattern. The same number of repetitions were used for all subsets. Identical pattern subsets were used as for the modulation of κ, along with the same values of α_0 in the second set of experiments.

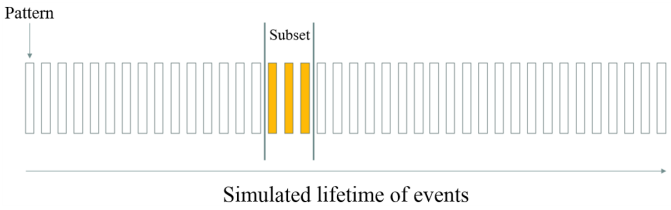


Fig. 8. A visual representation of the simulated lifetime. Each one of the 40 patterns resembles two years of episodic memories. The white bars resemble patterns which were trained without any extra modulation. The highlighted subset exemplifies one of the subsets used in which the patterns were trained with modulated plasticity.

E. Implementation

The network and its learning rules were implemented in Python using matrix and vector arithmetics. Euler's method was used in order to numerically approximate the differential equations present in Eq. 2-7. All simulations were run on computers with average performance levels.

IV. Results and Analysis

The purpose of the study was to demonstrate the effect on long-term memory recall of an additive momentary modulation of plasticity for a selected subset of memory patterns. This scenario was simulated in memory network models as presented in section III-B, following a training and recall paradigm as well as a testing paradigm described in detail in section III-C and III-D respectively. The subsections IV-A to IV-D each describe and analyze the results of the tests conducted.

A. Simulation of aging memories and base-case

To demonstrate the process of memories aging in the brain, one simulation was performed without including any subset of selectively modulated patterns. This then resembled the effect of decaying synaptic plasticity with age. In the simulation, training and recall was performed using the network parameter values listed in Tab. I and II. The result of this simulation clearly exhibits the phenomenon of reminiscence bump between the simulated ages of around ten to 30, occurring in the solid lines labeled "No mod." in Fig. 9 (B) and 10 (B). These simulation plots constitute reference curves, referred to as base-cases in the following sections. Both of the graphs also illustrate a small recency effect reflecting the network's preference for recently acquired memories. The two base-case plots differ slightly in shape even though they both were created using the simulation protocol in section III-C. This difference can be explained by both the randomness of pattern order and module deactivation, as well as the Gaussian noise added in Eq. 4, as all three of these factors are unique for every simulation.

B. Selective modulation in different simulated time periods

The encoding strength parameter, κ, was increased to 3 and selective plasticity modulation tests were performed as described in the testing paradigm, applying the increased value when training the specified subsets only. The results for all four of the selective modulations are presented in Fig. 9 (B)

TABLE IV
ACCUMULATED DIFFERENCES IN TEN YEAR BINS BETWEEN MODULATED GRAPHS AND BASE-CASE WHEN $\kappa = 3$

	Modulated subset			
Ages	**6-10**	**32-36**	**50-54**	**70-74**
2-10	0.115	0.120	0.055	0.0575
12-20	0.286	0.103	0.0575	0.0463
22-30	0.403	0.0933	0.0492	0.0367
32-40	0.416	0.0875	0.0419	0.0331
42-50	0.408	0.0995	0.0400	0.0320
52-60	0.398	0.109	0.0367	0.0304
62-70	0.385	0.110	0.0461	0.0332
72-80	0.389	0.112	0.0500	0.0328

TABLE V
NORMALIZED DIFFERENCES IN TEN YEAR BINS BETWEEN MODULATED GRAPHS AND BASE-CASE WHEN REPETITIONS = 3

	Modulated subset			
Ages	**6-10**	**32-36**	**50-54**	**70-74**
2-10	0.140	0.0825	0.0675	0.0400
12-20	0.298	0.550	0.588	0.400
22-30	0.405	0.0583	0.0458	0.0308
32-40	0.423	0.0613	0.0363	0.0269
42-50	0.411	0.0665	0.0310	0.0260
52-60	0.395	0.0763	0.0338	0.0250
62-70	0.383	0.0850	0.0346	0.0257
72-80	0.375	0.0838	0.0341	0.0278

together with the reference curve. In Tab. IV the difference between each graph and the base-case is normalized over ten year bins.

κ was reset to 1 and a modified number of repetitions, now increased to 3, was then introduced to modulate plasticity. New tests were performed according to the testing protocol with the same pattern subsets as used for the modulation of κ. The results of the recall ratio as a function of the simulated years are presented in Fig. 10 (B). Similarly to the modulation of κ, how much the graphs differ from the base-case was calculated and are listed in Tab. V.

The graphs representing the modulation of κ and number of repetitions presented in Fig. 9 (B) and 10 (B) all differed to varying degrees in comparison to the base-case. For both simulations it was clear that modulation of patterns situated later in the simulated lifetime resulted in a curve with greater resemblance to the base-case graph than for modulation of early situated patterns. This notion is supported by the difference calculations presented in Tab. IV and V, where the columns corresponding to later subsets show a smaller difference. The modulation of the patterns of corresponding ages six to ten years deviated the most from the base-case, clearly displaying a narrower "bump", as well as a rather low recall ratio for the succeeding ages. Another effect of early modulation was a clearly diminished recency effect. These results show that the modulation of a specific subset of patterns affects the recall of all other patterns and that this effect is more palpable for modulation of early situated subsets. The diminishing effect on all other recall attempts could suggest that the weights for the modulated subset of patterns outrival the weights for other patterns, making convergence to the modulated patterns more likely for all tested patterns.

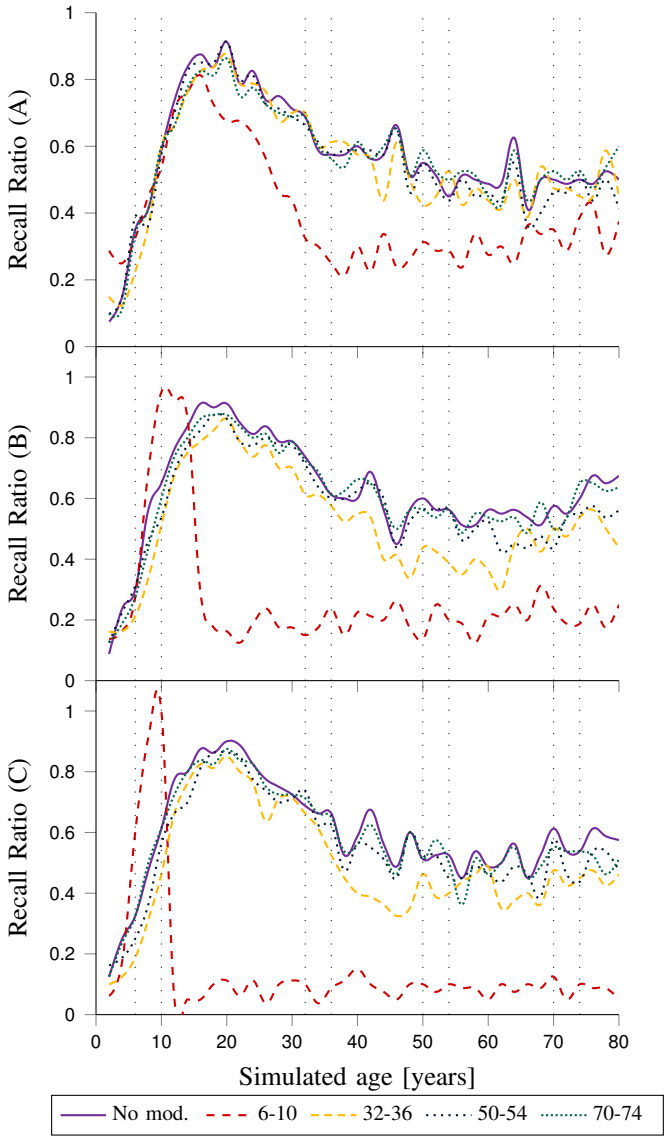


Fig. 9. Long-term memory recall with selective plasticity modulation implemented using an increased value of κ in regards to the base-case. Each line in the graphs refers to the modulated subset of patterns corresponding to the ages specified in the legend. The dotted vertical lines highlight the range of each modulated subset. "No mod." is short for no modulation and constitutes the base-case in which the value of $\kappa = 1$ for all memory patterns.
(A) Selective modulation applied to each represented subset using $\kappa = 2$
(B) Selective modulation applied to each represented subset using $\kappa = 3$
(C) Selective modulation applied to each represented subset using $\kappa = 4$

Additional tests would however been needed to be performed in order to support this hypothesis.

C. Varying encoding strength and number of repetitions

In order to determine the effect of selective plasticity modulations in terms of repetitions and encoding strength in learning, it was also necessary to study different values for the two modulation methods. Both modulation parameter values were assigned the values two and four respectively, and each value was tested using the same four subsets of patterns as in previous tests. From the results of these tests it was possible to determine that a higher value of κ and higher number of repetitions than used in the previous tests gave a more distinct

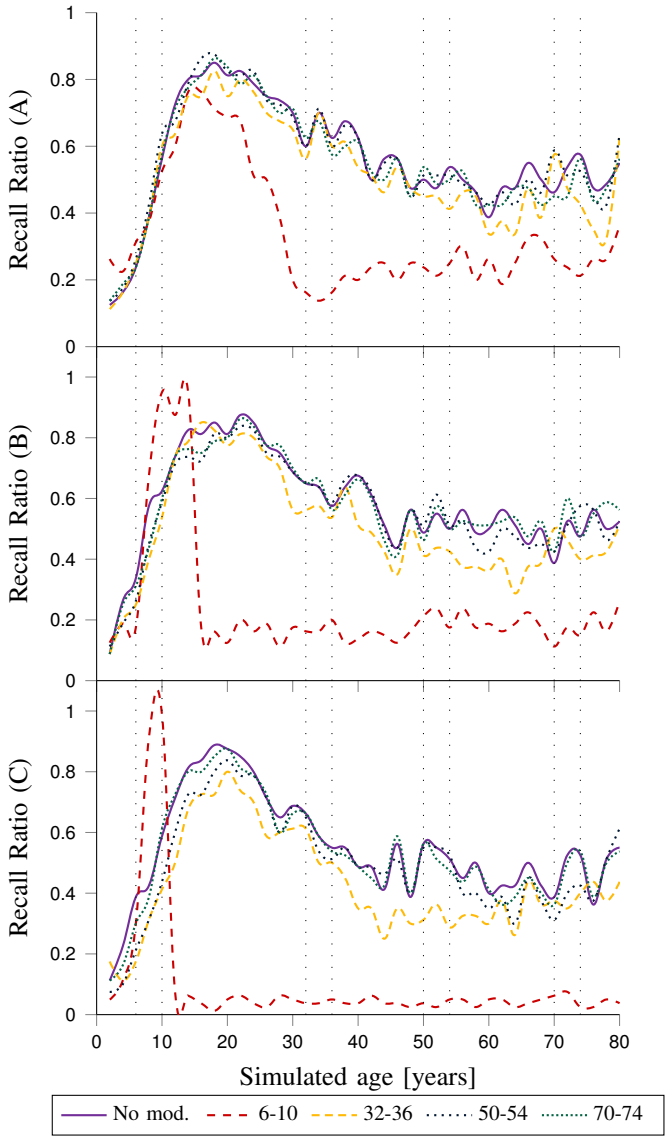


Fig. 10. Long-term memory recall with modulated repetitions for the four different subsets of patterns corresponding to the ages specified in the legend. Each line in the graph corresponds to one of the modulated subset. The dotted vertical lines highlight the range of each modulated subset. "No mod." is short for no modulation and constitutes the base-case in which the number of repetitions (rep.) $= 1$ was used for training all patterns.
(A) Selective modulation applied to each represented subset using rep. $= 2$
(B) Selective modulation applied to each represented subset using rep. $= 3$
(C) Selective modulation applied to each represented subset using rep. $= 4$

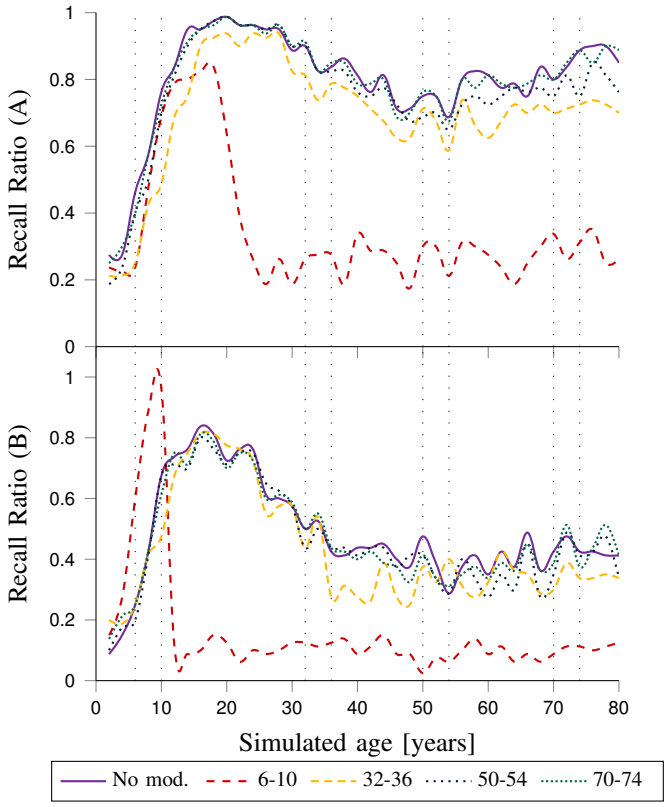


Fig. 11. Long-term memory recall with selective plasticity modulation implemented using an increased value of κ in tandem with two different values of α_0, 2.6 and 3.4 respectively. Each line in the graphs relates to one modulated subset corresponding to the ages specified in the legend. The dotted vertical lines highlight the range of each modulated subset. "No mod." is short for no modulation.
(A) Selective modulation applied to each represented subset using $\alpha_0 = 2.6$
(B) Selective modulation applied to each represented subset using $\alpha_0 = 3.4$

peak within the modulated subset. The results of using an increased parameter value also showed to impede the recall of the rest of the patterns more than for the two lower modulation values. Compare the dashed curves as a result of modulating the subset corresponding to the ages six to ten in Fig. 9 (A) and (C) for κ and Fig. 10 (A) and (C) for repetitions. When comparing the recency effects in all subplots in Fig. 9, it is clear that the effect diminishes with increased value of κ. The same observation can be made for increased number of repetitions when comparing the subplots in Fig. 10.

The plasticity gain, κ, affects the evolution of synaptic trace variables, while the repetition in learning modulates plasticity by iterating the training process for selected patterns. These two methods for modulating plasticity are in this aspect very different. The characteristics of the resulting graphs presented in Fig. 9 and 10 do however show that the two methods had similar effects on long-term memory recall, as curves corresponding to the same modulated subset in the graphs for the two methods have similar shapes.

D. Selective modulation in tandem with varying network parameter α_0

Another important aspect to study was the interplay between long-term and momentary plasticity modulation. The network's long-term plasticity was governed by α as described by Eq. 10, regulating the rate of plasticity decay over time. α itself was governed by the three parameters α_0, α_C, and τ_P. To vary the long term-plasticity, one therefore had to vary at least one of these parameters. In this experiment, α_0 was chosen as the varying parameter.

From the results of modulating κ along with a lowered value of $\alpha_0 = 2.6$ and an increased value of $\alpha_0 = 3.4$, it was possible to determine that for the lowest value of α_0, the bump itself was not as narrow when comparing with the graphs for the two higher values. Compare the graphs of modulation of the earliest pattern subset in Fig. 11 (A) and (B) where the

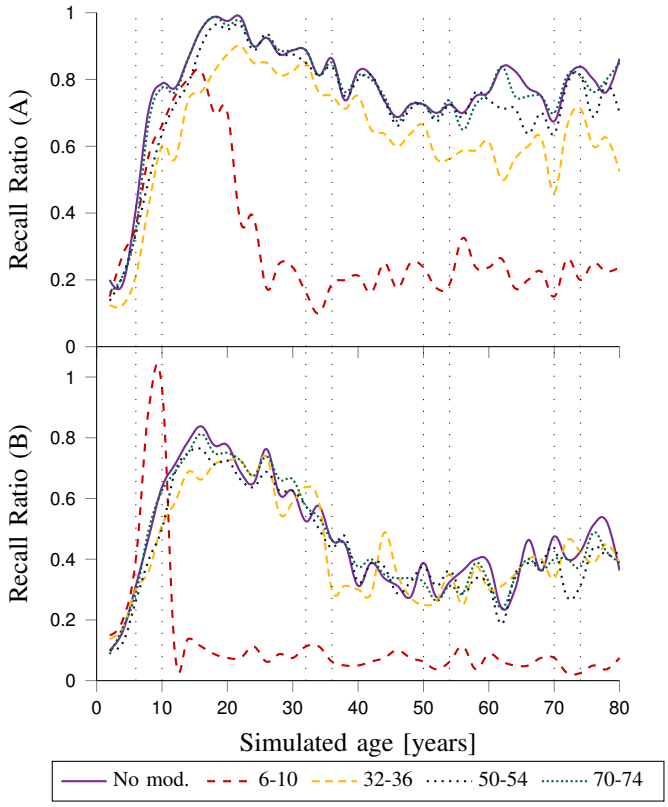


Fig. 12. Long-term memory recall with selective plasticity modulation implemented using an increased number of repetitions in tandem with two different values of α_0, 2.6 and 3.4 respectively. Each line in the graphs resembles one modulated subset corresponding to the ages specified in the legend. The dotted vertical lines highlight the range of each modulated subset. "No mod." is short for no modulation.
(A) Selective modulation applied to each represented subset using $\alpha_0 = 2.6$
(B) Selective modulation applied to each represented subset using $\alpha_0 = 3.4$

bump occurs between ages five and 25 for $\alpha_0 = 2.6$ and between two and twelve for $\alpha_0 = 3.4$. The overall recall ratio was also higher with decreasing values of α_0. In Fig. 11 (A), nearly every plot peaks at just under 1 in recall ratio, whereas in Fig. 11 (B) the majority of the plots peak at around 0.8. The same observations can be made in Fig. 12 (A) and (B) where equivalent tests for modulated repetitions in the same pattern subsets and with the same α_0-values were conducted.

V. Discussion

A. Summary of findings

The results of this study found that modulating the momentary plasticity of a long-term memory neural network model trained with the BCPNN learning rule had clear effects on recall capability. Modulation of patterns trained early in the simulated lifetime also had a larger impact on the overall recall than modulation of later patterns. The initial plasticity of the network was found to affect both the width and shape of the reminiscence bump in both cases of modulation. The two methods of modulating momentary plasticity through repetitions and varying the plasticity gain κ in training had similar effects on long-term memory recall.

B. Strengths and weaknesses of the study

One result obtained in this study was the emergence of a reminiscence bump due to the exponential decrease in network plasticity over time. The resulting graphs exhibit the phenomenon clearly but do all however have a relatively high recall ratio in comparison with theory [6]. This applies both in regards to the maximum value of the reminiscence bump as well as for the succeeding simulated ages. The curves representing the base-case, for example, reach a maximum value around 0.9. Since recall was performed once the network had trained all patterns, it would be possible to resemble this to the imaginary scenario of asking an 80-year-old person to recall a story of an event that occurred every year of the person's life. In this case, as the phenomenon of the reminiscence bump already explains, it is likely that the person will remember events from some years more clearly than others. It is however unlikely that the person will remember nearly every single detail of an event, even if it occurred during the period of time corresponding to the peak of the bump. From the perspective of the tests conducted in this study, the high values of recall ratio were still not considered to greatly impact the conclusions made. The focus was to evaluate the difference in recall ability due to plasticity modulation, rather than to analyze the recall statistics of each individual simulation. Nevertheless, the model as a whole would in general be considered more realistic from the perspective of human memory performance if this detail was addressed.

The results of the study also showed the tendency that a modulation of early subsets of patterns inhibited the characteristic of recency effect in long-term memory recall. This effect can be considered unrealistic from a biological perspective since it would imply, for example, that one would be unable to remember an event from the day before, just because of a strongly encoded memory of an event that occurred in their childhood. The effect displayed by the resulting graphs from the tests conducted in this study is however still seen as a tendency, since it also could be a result of added noise and an unfavourable contribution of randomness in the simulations. Multiple tests would have needed to be conducted in order to determine the cause of this effect.

Both due to limitations in time and in computing power, the network used consisted of five modules with ten units per module. Aside from the added Gaussian noise, this relatively small number of units could be one reason for in general noisy plots. If a larger network would have been used, together with an increased number of network simulations in total, then a more stable recall performance would have been expected. Since a rather simplistic method was used to calculate the differences in recall ratio between each graph and the base-case, the noisiness also affected these results. Other methods to compare the modulated and non-modulated graphs would therefore be desirable to further investigate.

If a greater number of simulations was used it would also have allowed for performing systematic statistical analysis which would have helped to distinguish any statistically significant effects. Again, this was not possible to do due to the restrictions of time caused by excessive computing time.

C. Cognitive hypotheses

When modulating the plasticity of the network, it is not explicitly defined in the model what the biological mechanisms underlying the modulation of plasticity are. In other words, there could be multiple different reasons for selective plasticity modulation in creation of certain memories. The proposed scenario of repeating inputs may naturally be compared with how learning is conducted in general forms of education. The modulation using an increased κ could however be likened to different factors.

Experiences associated with something highly stressful and/or emotional may generate memories that are stronger and better remembered [1] [27]. In regards to the modulation of the network parameter κ, it would therefore be possible to argue that this could resemble the process of creating a memory of an event that had some emotional or stressful association. As for its chemical equivalence in the brain, κ could be thought of as being a neuromodulator, which plays a part in upregulating neuronal plasticity in the brain [28]. If this was the case, then the modulation of κ could also reflect involuntary recall. This would mean that the memory with emotional value is more often involuntarily recalled than all other memories and could explain why all other memory patterns were affected. This explanation is strengthened by the fact that involuntary memories often are more emotionally weighted than voluntary memories [1].

It is known that it is possible to acquire effective methods of learning by simply wanting to learn [29]. Another hypothesis for the modulation of κ could therefore be that its equivalence in the brain would act as a primer, allowing the incoming input to be remembered well even though the input itself is not highly emotional or stressful. One example could be remembering the name of a person that one is introduced to. The name may be neutral in terms of emotional associations, but the importance of remembering the name itself could to some extent be emotionally relevant. This would then imply that the brain is primed to remember the incoming stimuli, independently of what the input turns out to be.

In both cases of selective plasticity modulation studied in this project, it would be possible to hypothesise that the modulation of plasticity corresponds to synaptic changes in the brain as a result of specific characteristics of the input. One could therefore reason that the recall of memories is not only dependent on the network parameters, or in reality the state of the brain, but also with which information and in what circumstances the brain is given memories to store.

The question of how the results of this study conforms to biology and cognitive psychology in reality is still left to be answered, seeing as no biological data has been found to compare the obtained results with. If similar results were to be found in future cognitive behavioural research, then it would imply that the characteristics of events and experiences themselves are important aspects to study in regards to memory recall. This may potentially provide an idea about how memories are stored as a result of different life experiences, without focusing on individual biochemical prerequisites. By gaining this knowledge, one could possibly use it to complement neurological studies where emotional experiences are of interest.

D. Future work

This study covers only two of many possible approaches to modulating plasticity selectively and examines only a single selected network plasticity parameter. Future work could further look into the impact of network parameters τ_P and α_C. Another possibility that could not be explored in this work due to time constraints was to study the impact of varying degrees of memory pattern overlap and how this would affect long-term memory recall. It would also be interesting to look into how prone all tested patterns were to converge to the modulated patterns.

VI. Conclusion

The experiments conducted in this study were aimed to answer the question of how selective BCPNN plasticity modulation affects long-term memory recall characteristics and effects such as reminiscence bump. The conclusion was that a selective modulation of plasticity when learning a subset of memory patterns promoted their successful recall in relation to all other patterns. This effect was overall more dramatic for memories encoded early in the simulated lifetime. Another finding was the similarity in the effects caused by the two variants of momentary plasticity modulation, despite their different methods of varying the plasticity in the network learning rule.

Acknowledgment

The authors would like to thank Pawel Herman for his invaluable guidance and support during this project, as well as for introducing the authors to the fascinating world of computational brain modeling. They would also like to thank Isabella Palmér for sharing her expertise in the field of medicine.

References

[1] D. C. Rubin, M. F. Dennis, and J. C. Beckham, "Autobiographical memory for stressful events: The role of autobiographical memory in posttraumatic stress disorder," *Conscious. Cogn.*, vol. 20, no. 3, pp. 840–856, Apr. 2011.

[2] D. Wang et.al , "Modelling autobiographical memory loss across life span," *AAAI-19*, vol. 33, no. 1, pp. 1368–1375, Jul. 2019.

[3] S. Janssen, A. Chessa, and J. Murre, "Modeling the reminiscence bump in autobiographical memory with the memory chain model," *Constructive memory*, pp. 138–47, Jul. 2003.

[4] M. Meeter and J. Murre, "Tracelink: A model of amnesia and consolidation." *Cogn. Neuropsychol.*, vol. 22, no. 5, pp. 559–587, Jul. 2005.

[5] Lansner lab KTH. (2021, Apr.) Bcpnn. [Online]. Available: https://www.csc.kth.se/forskning/cb/cbn/cbnweb.php?cont=bcpnn

[6] D. C. Rubin, T. A. Rahhal, and L. W. Poon, "Things learned in early adulthood are remembered best," *Mem. Cogn.*, vol. 26, pp. 3–19, Jun. 1998.

[7] K. Munawar, S. K. Kuhn, and S. Haque, "Understanding the reminiscence bump: A systematic review," *PLoS ONE*, vol. 13, no. 12, Dec. 2018.

[8] P. Herman, S. Benjaminsson, and A. Lansner, "Odor recognition in an attractor network model of the mammalian olfactory cortex," *IJCNN*, pp. 3561–3568, May 2017.

[9] N. B. Ravichandran, A. Lansner, and P. Herman, "Learning representations in bayesian confidence propagation neural networks," *IJCNN*, pp. 1–7, 2020.

[10] A. Lansner et.al, "Reactivation in working memory: An attractor network model of free recall," *PLoS ONE*, vol. 8, no. 8, Aug. 2013.

[11] P. Pereira, “Attractor neural network modelling of the lifespan retrieval curve,” Master thesis, KTH, Stockholm, Sweden, 2020.
[12] D. Purves et.al, “Synaptic plasticity” and ”Memory,” in *Neuroscience*, 5th ed. Cambridge, U.K. ; New York: Cambridge University Press, 2012, pp. 163–185, 695–715.
[13] W. Boron and E. Boulpaep, “Organization of the Nervous System,” in *Medical physiology*, 3rd ed. Philadelphia: Elsevier, 2017, pp. 269–271.
[14] D. O. Hebb, *The organization of behavior: A neuropsychological theory*. New York: John Wiley, 1949.
[15] Y. Munakata and J. Pfaffly, “Hebbian learning and development,” *Dev. Sci.*, vol. 7, no. 2, pp. 141–148, Apr. 2004.
[16] C. Keysers and V. Gazzola, “Hebbian learning and predictive mirror neurons for actions, sensations and emotions,” *Phil. Trans. R. Soc. B*, vol. 360, no. 1644, Jun. 2014.
[17] G. Miller. (2010, May) How our brains make memories. [Online]. Available: https://www.smithsonianmag.com/science-nature/how-our-brains-make-memories-14466850/
[18] T. Teyler and J. Rudy, “The hippocampal indexing theory and episodic memory: Updating the index,” *Hippocampus*, vol. 17, no. 12, pp. 1158–1169, Dec. 2007.
[19] M. A. Conway et.al, “A cross-cultural investigation of autobiographical memory: On the universality and cultural variation of the reminiscence bump,” *Cross-Cult. Psychol*, vol. 36, no. 6, pp. 739–749, Nov. 2005.
[20] T. Wolf and D. Zimprich, “What characterizes the reminiscence bump in autobiographical memory? new answers to an old question,” *Mem. Cogn.*, vol. 48, pp. 607–622, May 2020.
[21] H. Wang et.al., “Recurrent neural networks: Associative memory and optimization,” *J Inform Tech Soft Engg*, vol. 1, no. 2, pp. 1–15, Jan. 2011.
[22] C. Nicholson. (2021, Apr.) A beginner’s guide to neural networks and deep learning. [Online]. Available: https://wiki.pathmind.com/neural-network
[23] B. Olshausen, “Bayesian probability theory,” *RCTN*, pp. 1–6, Jan. 2004.
[24] A. Sandberg, A. Lansner, K. M. Petersson, and Ö. Ekeberg, “A bayesian attractor network with incremental learning,” *Network*, vol. 13, no. 2, pp. 179–194, Jun. 2002.
[25] C. Johansson, A. Sandberg, and A. Lansner, “A capacity study of a bayesian neural network with hypercolumns,” *internal report, KTH, Stockholm*, Apr. 2012.
[26] SCB. (2021, Mar.) Medellivslängden i Sverige. [Online]. Available: https://www.scb.se/hitta-statistik/sverige-i-siffror/manniskorna-i-sverige/medellivslangd-i-sverige/
[27] J. Quevedo et.al, “Differential effects of emotional arousal in short- and long-term memory in healthy adults,” *Neurobiol. Learn. Mem.*, vol. 79, no. 2, pp. 132–135, Mar. 2003.
[28] V. Pedrosa and C. Clopath, “The role of neuromodulators in cortical plasticity. a computational perspective,” *Front. Synaptic Neurosci.*, vol. 8, Jan. 2017.
[29] J. Li, “A cultural model of learning: Chinese “heart and mind for wanting to learn”,” *J Cross Cult Psychol*, vol. 33, no. 3, pp. 248–269, May 2002.

Effects of Network Size in a Recurrent Bayesian Confidence Propagating Neural Network With two Synaptic Traces

Ludwig Karlsson and William Laius Lundgren

***Abstract*—A modular Recurrent Bayesian Confidence Propagating Neural Networks (BCPNN) with two synaptic time traces is a computational neural network that can serve as a model of biological short term memory. The units in the network are grouped into modules called hypercolumns within which there is a competitive winner-takes-all mechanism.**

In this work, the network's capacity to store sequential memories is investigated while varying the size of and number of hyperocolumns in the network. The network is trained on sets of temporal sequences where each sequence consist of a set of symbols represented as semi-stable attractor state patterns in the network and evaluated by its ability to later recall the sequences.

For a given distribution of training sequence the networks' ability to store and recall sequences was seen to significantly increase with the size of the hypercolumns. As the number of hypercolumns was increased, the storage capacity increased up to a clear level in most cases. After this point it was observed to remain constant and did not improve by adding any more hypercolumns (for a given sequence distribution). The storage capacity was also seen to depend a lot on the distribution of the sequences.

***Sammanfattning*—Ett modulärt *Recurrent Bayesian Confidence Propagating Neural Network* (BCPNN) med två synaptiska tidsspår är ett neuronnät som kan användas som en modell för biologiskt korttidsminne. Enheterna i nätverket är grupperade i moduler kallade hyperkolumner inom vilka enheterna konkurrerar enligt en "winner-takes-all"-mekanism.**

I det här arbetet undersöktes hur nätverkets förmåga att lagra sekventiella minnen beror på storleken och antalet hyperkolumner. Nätverket tränades på ett antal temporala följder där varje följd bestod av en mängd symboler som representerade som attraktor-tillstånd i nätverket och bedömdes baserat på dess förmåga att komma ihåg följder det lärt sig under träning.

För en given fördelning av träningsföljder ökade nätverkets förmåga att lagra och återkalla följder med storleken på hyperkolumnerna. Då antalet hyperkolumner ökades ökade också i de flesta fall lagringsförmågan upp till en viss nivå varefter ytterligare hyperkolumner inte gav några vidare förbättringar (för en given fördelning av sekvenser). Lagringskapaciteten berodde också mycket på fördelningen av följder.

***Index Terms*—BCPNN, computational brain modelling, short-term memory, sequential learning, hypercolumns.**

Supervisors: Pawel Herman

TRITA number: TRITA-EECS-EX-2021:192

I. Introduction

A. Background

As we all experience, biological brains are capable of learning a large amount of sequential information. During memory recall in biological brains, different particular assemblies of neurons are active in a sequential order [1]. Short-term sequential memory in humans allows for the real time recognition and disambiguation of temporal sequences[1] acquired through our senses such as vision [2].

Donald Hebb theorized that neurons which fire together tend to have stronger connections. This has become known as *Hebb's Rule* [3]. *Hebbian learning* is a learning scheme which takes inspiration from Hebb's rule, and can be used for unsupervised learning in artificial neural networks. One way to implement Hebbian learning is to use Bayesian statistics to determine the strength of the connections between two neurons based on the probability of them being stimulated simultaneosly [3]. It can be used to train *recurrent associative networks*, in which signals are recurrently sent between units until the network converges to certain *attractor states*[2], causing the network to associate input patterns with attractor states [3]. If asymmetric connections between units are used, the network can associate input with a temporal sequence of semi-stable attractor states. This forms the basis of the BCPNN (Bayesian Confidence Propagating Neural Network) learning rule described in section II (Method) and used in this work. This model captures the previously discussed aspects of short-term memory well [4] [5].

When learning sequences, a network needs some way of keeping track of recent activity, not just present activity. One mechanism which allows the network to do this is *synaptic time traces*. A synaptic[3] time trace refers to having the previous activation history of a neuron affect its current state, and in these models two time traces—one with a longer time span and one with a shorter time span—were used. Two such time traces are also seen in the firing dynamics of biological neurons [6]. Recently, there has been research done on neural networks modelling biological short-term memory sequence recognition with a recurrent BCPNN with two synaptic time traces [7] and neurons grouped into modules called hypercolumns[4]. Using one synaptic time trace was also tested, but adding a second

[1]Temporal sequences refers to an ordered sequence of symbols appearing after each other in time.

[2]An attractor state is a state that the network tends to converge to over time if the starting state and input lies within a specific region. It can be physically interpreted as a potential energy minimum.

[3]They are called synaptic since they measure in- and out-signals of the neurons which are conveyed by their synapses (or analogously connections in the implementation).

[4]Hypercolumns in the case of this report is simply the name of the groups of units within which there is a winner-takes-all mechanism

facilitated sequence disambiguation [8] [9]. In the previous studies the network has been used to model various aspects of memory, but the focus has not been on the ability of this model to learn larger amounts of sequential information when the network is scaled up (i.e. increasing the number of units and hypercolumns in the network) [8] [7] [9]. This project aims to investigate how the performance of this network to store and recall sequential memories depends on hypercolumn size and number of hypercolumns. In addition to providing insight into the properties of this model of short term memory, the research could be valuable from a brain-like computing perspective as it paves the way for the development of brain-inspired memory systems with unique computing capabilities.

B. Research Question

Considering a modular recurrent BCPNN with two synaptic time traces, how does the number of partially overlapping sequences from a given distribution that the network can learn reliably depend on the size and structure of the network?

C. Scope

In this work, focus is on how the storage capacity of the network qualitatively changes with the size and shape of the network. As such the other network hyperparameters were not optimized to maximise storage. Many hyperparameter choices were taken from previous research and not investigated further on the assumption that small changes would not drastically change the qualitative behavior of the results. (An exception was made for a particular parameter, the adaptation gain.)

II. Method

A. Method Overview

The network's structure is described in subsection II-B and its dynamics is governed by a system of differential equation described in subsection II-C. The equations were numerically integrated using forward Euler with time step Δt.

The network was trained and subsequently tested on a few different sets of randomly generated integer sequences (described in section II-E) using training and testing algorithms described in detail in sections II-D and II-H. The integers are represented in the network as activation patterns which act as semi-stable attractor states, and the mapping from integers to patterns in the network is described in sections II-F and II-G. Parametric studies for some network hyperparameters modulating the size and configuration of the network were performed, and for each parameter choice the network was trained and tested many times to produce statistical data. The network that was investigated is based on the similar BCPNN model used in [7], which in turn builds on the previous work in [8] and [9].

A single instance of the simulation with a particular set of parameter choices is referred to as a *case*. A set of instances with the same parameters is referred to as a *batch* with a particular *batch size*. A set of batches constitutes a *study*. The program flow for a single case is described in figure 1. This system was integrated in a test setup that allowed the researchers to stage and run large sets of predetermined cases in order to generate statistical data for different choices of hyperparameters and randomized input data. Section II-I lists the ranges and choices of hyperparameter investigated in this work, and section II-J defines the performance metrics used to evaluate the networks.

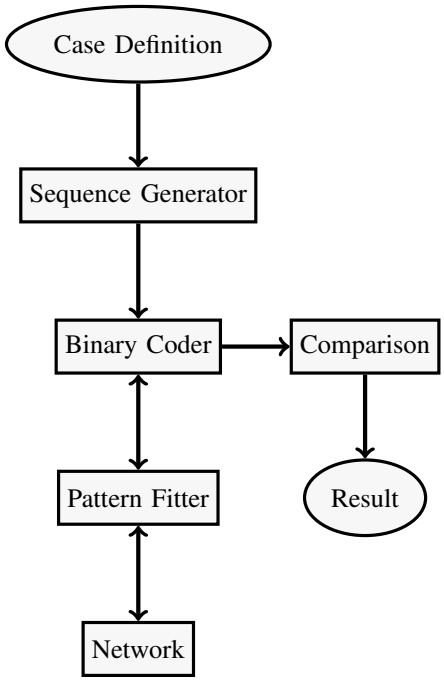


Fig. 1. Program flow for a single case simulation. The *Sequence Generator* (see section II-E) generates a set of sequences of integers based on the *Case Definition*. The *Binary Coder* (see section II-F) then encodes the sequences' integers into binary representations. The *Pattern Fitter* (see section II-G) takes the binary representations to patterns that fit the current network structure. These lists of patterns are then fed as training data to the *Network* (see sections II-B, II-C, and II-D). During recall, patterns read from the network (see section II-H) are fed back to the Pattern Fitter which deconstruct them into binary codes. The binary codes are further decoded by the Binary Coder which results in integers (symbols decoded from the network) stored in a *recalled sequence*. These recalled sequences are finally compared to the original sequences. A success fraction is determined based on how many sequences that were correctly recalled. This constitutes the *Result* of the case simulation.

B. Network Structure

The network consists of H hypercolumns each containing $N = 2^k$ (for some $k \in \mathbb{N}$) units[5]. Within each hypercolumn there is a winner-takes-all (WTA) mechanism, such that the unit i with the highest activity s_i in its hypercolumn has its binary activation o_i set to 1 while the others have their binary activations set to $o_i = 0$ (see equation (6)). There are connections between all units in the network, with strengths of the connections determined by training. The connections can be either inhibitory or excitatory[6]. The network structure is visualized in figure 2.

[5]The limitation of the size of the hypercolumns to powers of two ($N = 2^k$) comes from how symbols are represented in the network, as further described in section II-G.

[6]A connection from unit A to unit B is inhibitory if activity in A reduces activity in B, and excitatory if activity in A increases activity in B.

C. Network Dynamics

In the vector $\mathbf{s}$, element s_i is the activity of unit i. $\mathbf{s}$ is updated according to equation (1),

$$\tau_s \frac{d\mathbf{s}}{dt} = \boldsymbol{\beta}^{\text{fast}} + \boldsymbol{\beta}^{\text{slow}} + \mathbf{W}^{\text{fast}}\mathbf{z}_{\text{pre}}^{\text{fast}} + \mathbf{W}^{\text{slow}}\mathbf{z}_{\text{pre}}^{\text{slow}} - g_a\mathbf{a} - \mathbf{s} + \mathbf{I} \quad (1)$$

where τ_s and g_a are time constants, $\mathbf{I}$ is an externally applied current, and $\mathbf{a}$ is the *adaptation* governed by (7). The vectors $\boldsymbol{\beta}^{\text{fast}}$ and $\boldsymbol{\beta}^{\text{slow}}$ are biases while the matrices $\mathbf{W}^{\text{fast}}$ and $\mathbf{W}^{\text{slow}}$ are the weight matrices (calculated during training using fast and slow time traces respectively, described in section II-D: *Learning rule*).

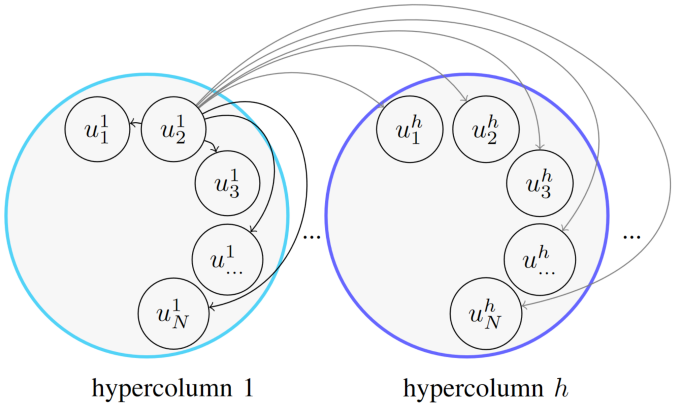


Fig. 2. Schematic diagram of network structure showing hypercolumn 1 (light blue) and hypercolumn $h \leq H$ (dark blue) with their N units each. The connection from unit 2 in hypercolumn 1 to all other units in hypercolumn 1 (black) as well as the connections from unit 2 in hypercolumn 1 to all units in hypercolumn h (gray) are drawn. An equally large set of connections will stem from every unit, but the rest are omitted from the figure for clarity.

The variable vectors $\mathbf{z}_{\text{pre}}^{\text{slow}}$, $\mathbf{z}_{\text{post}}^{\text{slow}}$, $\mathbf{z}_{\text{pre}}^{\text{fast}}$, and $\mathbf{z}_{\text{post}}^{\text{fast}}$ are the fast-changing and slow-changing pre-synaptic and post-synaptic activity time traces, which are updated according to equations (2)-(5):

$$\tau_{z_{\text{pre}}^{\text{slow}}} \frac{d\mathbf{z}_{\text{pre}}^{\text{slow}}}{dt} = \mathbf{o} - \mathbf{z}_{\text{pre}}^{\text{slow}} \quad (2)$$

$$\tau_{z_{\text{post}}^{\text{slow}}} \frac{d\mathbf{z}_{\text{post}}^{\text{slow}}}{dt} = \mathbf{o} - \mathbf{z}_{\text{post}}^{\text{slow}} \quad (3)$$

$$\tau_{z_{\text{pre}}^{\text{fast}}} \frac{d\mathbf{z}_{\text{pre}}^{\text{fast}}}{dt} = \mathbf{o} - \mathbf{z}_{\text{pre}}^{\text{fast}} \quad (4)$$

$$\tau_{z_{\text{post}}^{\text{fast}}} \frac{d\mathbf{z}_{\text{post}}^{\text{fast}}}{dt} = \mathbf{o} - \mathbf{z}_{\text{post}}^{\text{fast}} \quad (5)$$

where o_i is 1 if s_i has the maximal value in its hypercolumn, and 0 otherwise according to the winner-takes-all mechanism. This is described more precisely by equation (6) which specifies $\mathbf{o}$'s value for the n:th unit of the h:th hypercolumn. Consequently, H of the $H \cdot N$ units in $\mathbf{o}$ will be 1 at a given moment.

$$o_{h+n} = \begin{cases} 1 \text{ if } s_{h+n} \text{ is maximal in hypercolumn } h \\ 0 \text{ otherwise} \end{cases} \quad (6)$$

The adaptation $\mathbf{a}$ is updated according to equation (7):

$$\tau_a \frac{d\mathbf{a}}{dt} = \mathbf{o} - \mathbf{a} \ . \quad (7)$$

To display the dynamics of the network during sequence recall, the successful behavior of a network with a single hypercolumn trained on a single sequence is explored in detail in figures 3 and 4.

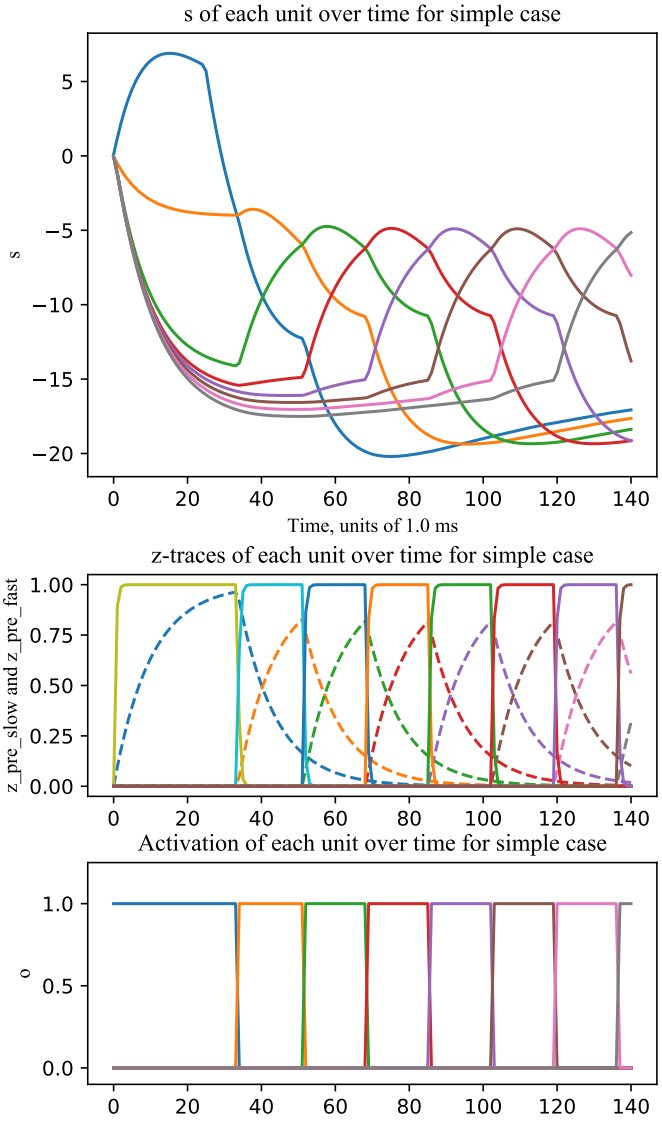


Fig. 3. The behavior of a small network during sequence recall. A simple case with $|S| = 8$, $L_s = 8$, $N_s = 1$, $H = 1$, $N = 8$, $g_a = 14$ was used. The sequence was manually selected to be (0,1,2,3,4,5,6,7). **The top plot** shows the development of the components of the activity vector $\mathbf{s}$ over time. Each line represents one unit. The high value of the first unit's activity is due to that unit being clamped with the external current I at the beginning of sequence recognition. This initiates the proceeding domino-like effect where the sequence's symbols' patterns are activated in order. The simulation is terminated when the final symbol has been recalled. **The middle plot** similarly displays the presynaptic $\mathbf{z}$-traces. The dotted lines corresponds to the slow trace and the solid line corresponds to the fast trace. **The bottom plot** shows the development of $\mathbf{o}$ over time. The behavior of this plot clearly shows the WTA mechanism in action.

D. Learning Rule

The weight matrices $\mathbf{W}^{\text{slow}}$, $\mathbf{W}^{\text{fast}}$ and the bias vectors $\boldsymbol{\beta}^{\text{slow}}$, $\boldsymbol{\beta}^{\text{fast}}$ are calculated using the probability traces $\mathbf{p}_{\text{pre}}^{\text{slow}}$, $\mathbf{p}_{\text{post}}^{\text{slow}}$, $\mathbf{p}_{\text{joint}}^{\text{slow}}$, $\mathbf{p}_{\text{pre}}^{\text{fast}}$, $\mathbf{p}_{\text{post}}^{\text{fast}}$, and $\mathbf{p}_{\text{joint}}^{\text{fast}}$ which are procedurally updated during training according to equations (8)-(13):

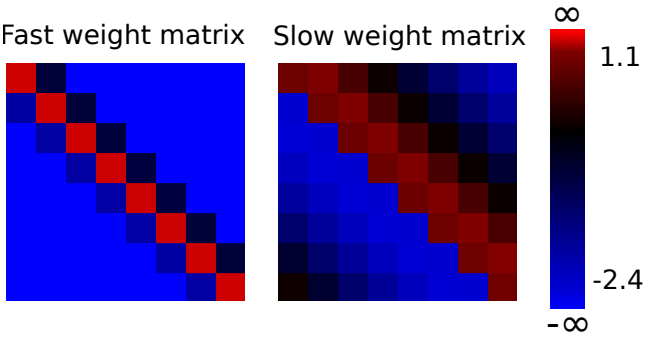


Fig. 4. The weight matrices for the network displayed in figure 3. A simple case with $|S| = 8$, $L_s = 8$, $N_s = 1$, $H = 1$, $N = 8$, $g_a = 14$ was used. The sequence was manually selected to be (0,1,2,3,4,5,6,7). Notice how units have positive (excitatory) connections to the next unit in the sequence, while they have negative (inhibitory) connections to the previous units. In the slow matrix this persists decreasingly for units farther away in the sequence as well. (The fast weight matrix contains values between -11.5 and 2.3 while the slow weight matrix contains values between -2.4 and 1.1 according to the color scale (the color scale is sigmoidal in the weights).)

$$\tau_p \frac{d\mathbf{p}_{\text{pre}}^{\text{slow}}}{dt} = \mathbf{z}_{\text{pre}}^{\text{slow}} - \mathbf{p}_{\text{pre}}^{\text{slow}} \tag{8}$$

$$\tau_p \frac{d\mathbf{p}_{\text{post}}^{\text{slow}}}{dt} = \mathbf{z}_{\text{post}}^{\text{slow}} - \mathbf{p}_{\text{post}}^{\text{slow}} \tag{9}$$

$$\tau_p \frac{d\mathbf{p}_{\text{joint}}^{\text{slow}}}{dt} = \mathbf{z}_{\text{pre}}^{\text{slow}} (\mathbf{z}_{\text{post}}^{\text{slow}})^{\text{T}} - \mathbf{p}_{\text{joint}}^{\text{slow}} \tag{10}$$

$$\tau_p \frac{d\mathbf{p}_{\text{pre}}^{\text{fast}}}{dt} = \mathbf{z}_{\text{pre}}^{\text{fast}} - \mathbf{p}_{\text{pre}}^{\text{fast}} \tag{11}$$

$$\tau_p \frac{d\mathbf{p}_{\text{post}}^{\text{fast}}}{dt} = \mathbf{z}_{\text{post}}^{\text{fast}} - \mathbf{p}_{\text{post}}^{\text{fast}} \tag{12}$$

$$\tau_p \frac{d\mathbf{p}_{\text{joint}}^{\text{fast}}}{dt} = \mathbf{z}_{\text{pre}}^{\text{fast}} (\mathbf{z}_{\text{post}}^{\text{fast}})^{\text{T}} - \mathbf{p}_{\text{joint}}^{\text{fast}} \ . \tag{13}$$

The $\mathbf{z}$-traces are also updated during training according to equations (2)-(5). The list of training sequences are repeated E_n times, each of which is called an epoch. The final values of the probability traces are attained after the list of training sequences have been iterated through E_n times. From these final values, the weight matrices and bias vectors are calculated according to (14)-(17) [7]:

$$(W^{\text{slow}})_{i,j} = \ln\left(\frac{(p_{\text{joint}}^{\text{slow}})_{i,j}}{(p_{\text{pre}}^{\text{slow}})_i \cdot (p_{\text{post}}^{\text{slow}})_j}\right) \tag{14}$$

$$(\beta^{\text{slow}})_i = \ln\left((p_{\text{post}}^{\text{slow}})_i\right) \tag{15}$$

[7]Note that the natural logarithm was used here, while previous research done by [7], [8], [9] used the base 10 logarithm. In practice this introduces a constant factor on the weights and biases in the equations which could have some effect on parameter choices and the behavior of the network.

$$(W^{\text{fast}})_{i,j} = \ln\left(\frac{(p_{\text{joint}}^{\text{fast}})_{i,j}}{(p_{\text{pre}}^{\text{fast}})_i \cdot (p_{\text{post}}^{\text{fast}})_j}\right) \tag{16}$$

$$(\beta^{\text{fast}})_i = \ln\left((p_{\text{post}}^{\text{fast}})_i\right) \ . \tag{17}$$

When the network is trained on a sequence, all the symbols in the sequence are iterated over, with the program setting $\mathbf{I}$ to the pattern representing the current symbol in the sequence (see subsections II.F and II.G) forcing that pattern to activate, and waiting for T_P before updating it to the pattern corresponding to the next symbol. Between sequences, the program waits an additional time ISI (the *Inter Sequence Interval*).

E. Sequence Generators

Since disambiguation of ambiguous input is an important feature of biological sequential memory [10], the network was trained and tested on sets of sequences sharing common symbols with each other—this is referred to as *sequential overlap*.

Sequences were drawn from a two different distributions, with their own generators. A sequence generator generates a set of sequences according to a distribution described by the following parameters: sequence length L_s, number of sequences N_s, the overlap rate R_r, and the set S of symbols that can be used in the sequences. The sequence families that were used are:

I. Sequences with no overlap.
II. Sequences with no overlap in the beginning and the end (first two slots and last slot), and a randomly generated subsequence the in the middle with statistically expected overlap rate R_r.

The algorithms used to generate them are described in detail below.

Generator I: Generates a set $\{\text{seq}_1, \text{seq}_2, ..., \text{seq}_{N_s}\}$ of N_s sequences with length L_s consisting of symbols randomly drawn from S with the condition that no symbol can be used more than once in the sequence set.

Generator II: Generates a set $\{\text{seq}_1, \text{seq}_2, ..., \text{seq}_{N_s}\}$ of N_s sequences with length L_s. A set S_{unique} containing $3L_s$ symbols randomly drawn from S is generated. The elements of S_{unique} are then assigned randomly to the first two slots in every sequence and last slot in every sequence in such a way that no symbol in S_{unique} is used more than once in the sequence set.

A fraction R_r of the elements in $S \setminus S_{\text{unique}}$ are randomly selected to appear twice in a new list S_{new}, while the rest of the elements in $S \setminus S_{\text{unique}}$ appear once in S_{new} (together, these constitute all the elements in S_{new}). The symbols to fill the middle sections of all sequences are then randomly drawn without replacement from S_{new} with the condition that the same symbol may not appear in two adjacent sequence slots. Note that the actual ratio of overlap, meaning the probability that an arbitrary symbol in the middle sub-sequence part of the sequence is not unique, will be $R_o = \frac{2R_r}{1+R_r}$.

In this work, integers were used as symbols. This was done because using integers as symbols makes it easy to reason about the sequences. It is also a natural choice from

an implementation perspective. Doing this does not limit the generality since the integers could easily be replaced with any other finite set through a bijection. Alternatively, the integers could be seen as an enumeration of a subset of patterns.

F. Encoder

The encoder encodes integer symbols into binary representations in the form of binary vectors. The elements of the binary vector are simply the digits of the binary representation of the integer in question. A common vector size D of the binary representations is determined based on the symbol set size $|S|$ such that

$$D = \lceil \log_2 |S| \rceil . \tag{18}$$

G. Pattern Fitter

The *Pattern Fitter* serves two purposes: it maps binary vectors onto activation patterns and it deconstructs activation patterns into binary vectors. This makes it possible to go back and forth between binary representations of symbols and the network's representations of those same symbols. In order for this to work at all, the number of realizable patterns in the network needs to be at least as many as the 2^D possible input vectors.

Due to the WTA mechanism, each hypercolumn may at any time occupy one of N possible activation states. If N is chosen to be 2^k, then all of a hypercolumn's states can be enumerated with all of the binary numbers of k digits[8]. If the hypercolumns state numbers are concatenated, all binary numbers of Hk digits represent all possible activation states of the network in this way.

In order to map the binary symbol codes of length D to these binary network activation state representations of length Hk, the binary code is repeated as to construct a binary number with the same number of digits as the network representations. When instead mapping in the other direction and deconstructing a state representation into a binary vector, the symbol who's network state representation is closest to the current state representation in terms of Hamming distance is chosen[9]. Figure 5 gives an example of this process.

If the size $|S|$ of the symbol set is greater the number N of unique activation patterns in a hypercolumn, several symbol representations need to share the same patterns in some hypercolumns. This is referred to as *spatial overlap*.

This Pattern Fitter together with some encoder gives an easy to implement, modular, and deterministic way to assign activation states to symbols and vice versa. If the network is sufficiently large, it also avoids two symbols representations being very similar in a sense of having many shared hypercolumn states[10].

[8]For example if $k = 3$, a hypercolumn may be in state $\mathbf{o} = (0, 0, 0, 0, 1, 0, 0, 0)$ which can be represented by the binary number 100

[9]Depending on the size of the network there may not be an unambiguous closest fit. One way to resolve these ties (which was used here) is to determine the closest symbol code with a bitwise majority gate which bit-wise evaluate ties to 0. This gate is applied to all of the "same" digit in all repetitions and then determines the resulting digit based on which digit is most common.

[10]For example, with $D = 4$, $H = 6$, and $k = 3$, all symbol representations are guaranteed to differ in at least four hypercolumns.

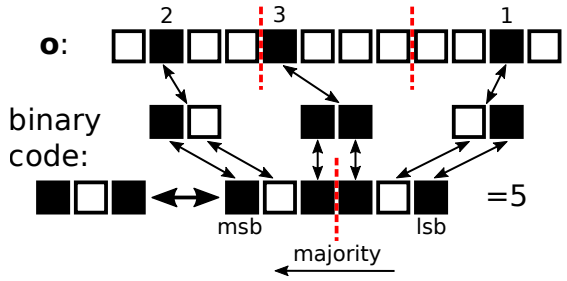


Fig. 5. The Pattern Fitter takes a binary representation of a symbol with D digits and repeats it to create a binary number with Hk digits. This number is split up into H parts which determine which unit is active in each hypercolumn. To deconstruct a pattern it works similarly except that it uses a bit-wise majority gate to resolve any inconsistencies. In this example the number five represented by a three digit number ($D = 3$) is fit to a network with three hypercolumns ($H = 2$) of dimension two ($k = 2, N = 2^2 = 4$). In the figure binary numbers are represented with the most significant bit in the leftmost position, and with black squares representing high bits.

TABLE I
HYPERPARAMETER CHOICES: THE FIRST SECTION OF THE TABLE CONTAINS THE HYPERPARAMETERS SPECIFIYING THE NETWORK MECHANICS IN THE DIFFERENTIAL EQUATIONS (1)-(5), AND (8)-(13). THE SECOND SECTION CONTAIN THE HYPERPARAMETERS SPECIFYING THE NETWORK SIZE. THE THIRD SECTION CONTAINS ADDITIONAL HYPERPARAMETERS SPECIFYING THE EXTERNAL STIMULI DURING TRAINING OF THE NETWORK. THE FORTH SECTION FINALLY LISTS THE HYPERPARAMETERS SPECIFYING THE TESTING OF THE NETWORK.

Symbol	Parameter Name	value
g_a	Adaptation gain	$8, (4-20)$
τ_a	Adaptation time constant	250 ms
τ_s	Unit time constant	10 ms
$\tau_{z_{\text{pre}}^{\text{slow}}}$	Slow pre-synaptic trace time constant	100 ms
$\tau_{z_{\text{post}}^{\text{slow}}}$	Slow post-synaptic trace time constant	5 ms
$\tau_{z_{\text{pre}}^{\text{fast}}}$	Fast pre-synaptic trace time constant	5 ms
$\tau_{z_{\text{post}}^{\text{fast}}}$	Fast post-synaptic trace time constant	3 ms
τ_p	Probability time constant	5000 ms
Δt	Time step	0.1 ms
N	Number of units per hypercolumn	2^k
k	Hyper column dimension	$3-6$
H	Number of hypercolumns	$2-10$
T_P	Pulse time	50 ms
E_N	Number of training epochs	50
ISI	Inter sequence interval	100 ms
T_C	Clamp time	25 ms
T_{set}	Settling time	4 ms
I_{scale}	I-scale factor	15

H. Testing Details

When testing the network's ability to recall a sequence, an external current $\mathbf{I}$ corresponding to the activation pattern $\mathbf{o}$ related to the first symbol in the sequence (see subsections II.F, II.G) is applied for a time T_C, with the current's amplitude scaled to I_{scale}. Because of the dynamics of the network, this sets a series of new patterns in motion. When a pattern change (a change in $\mathbf{o}$) has persisted for longer than a time T_{set}, $\mathbf{o}$ is read and its corresponding symbol is interpreted as the next recalled symbol.

I. Hyperparameter Choices

A list of all the hyperparameters used to describe the network is given in Table I. Fixed parameter choices are informed

by a combination of parameter studies and exploration in section III (RESULT) as well as previous research. The starting values of the hyperparameter choices were based on the values and intervals of the hyperparameters used in [7].

J. Definition of Maximum Storage Capacity, Storage Capacity, and Success Rate

In section III (RESULTS) the *storage capacity* (SC) of the network is used as a measure of the amount of sequential information that the network can learn. It if defined as

$$SC = L_s \cdot N_s \cdot R_s \tag{19}$$

where R_s is the *success fraction*, which is the fraction between the number of sequences that were correctly recalled and the total number of sequences, during the testing of the network for a given case. Storage capacity was used since it measures the total number of symbols that the network can store in a way that is agnostic to the lengths of the sequences. For example, storing four sequences of length five and two sequences of length ten gives the same SC.

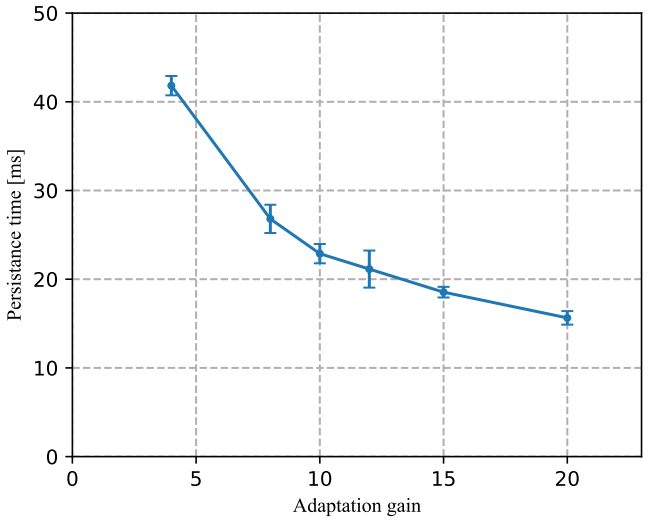


Fig. 6. Average persistence time $\bar{T}_{\text{per}}$ plotted against adaptation gain g_a. The measurements were made in a network with the same structure as that in figure 7: $H = 6$ and $N = 2^5 = 32$. In each experiment the network recalled three sequences. In total, averages are calculated from non-initial symbols during recall of 15 sequences of length 10. The error bars are 95% confidence intervals using the estimated standard error.

III. RESULTS AND ANALYSIS

A. Verification of Network Function and Initial Parameter Choices

In order to make some final decisions on parameter choices, two parameter studies on the adaptation gain g_a were performed. This was because previous research did not present a specific value (but a broader range of values) for g_a [7] [8] [9], and we found that performance varied for g_a values within those ranges. The first study looked at three widely distributed values for the adaptation gain: $g_a \in \{4, 12, 20\}$. While the network performed comparably bad in the case of $g_a = 4$ (see figure 7), no significant difference in the performance could be seen between $g_a = 12$ and $g_a = 20$. A second study looked closer at the range around $g_a = 12$ which seemed most promising, by investigating $g_a \in \{8, 10, 12, 15\}$ (see figure 7, plot B). No significant difference in the performance could be seen between the different values.

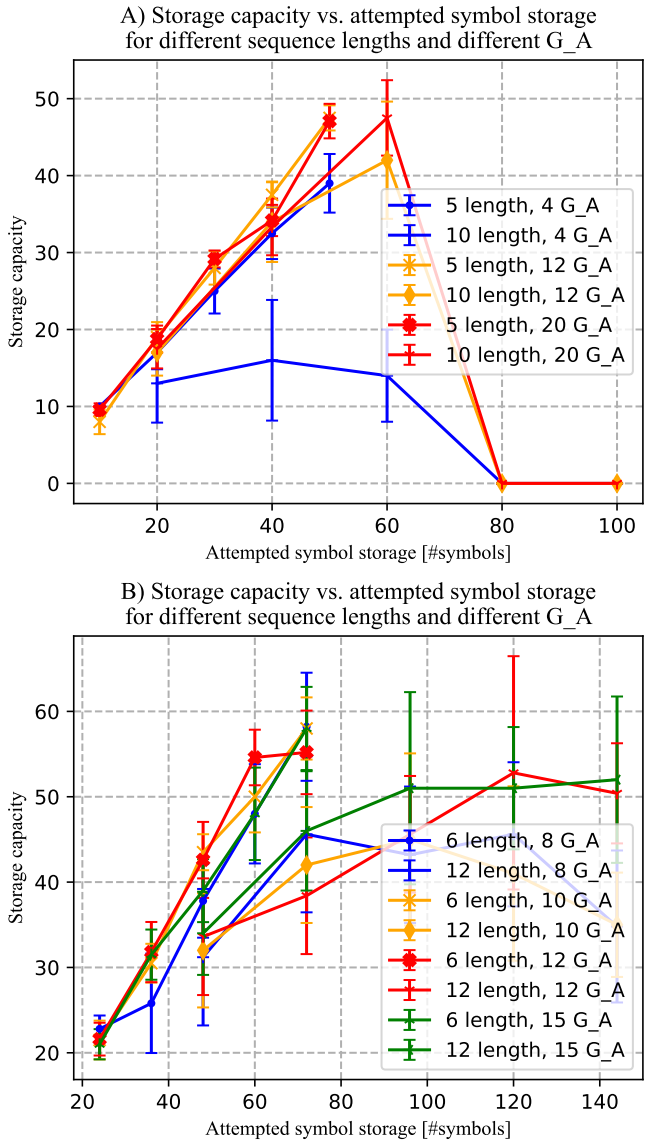


Fig. 7. The storage capacity of the network's dependence on the adaptation gain g_a for a few different sequence lengths L_s. Plot **A)** shows a results from experiments where a wide spread of $g_a = 4, 12, 20$ was tested. Here slightly shorter sequences and $|S| = 64$ was used. While all networks performed comparably well for shorter sequences, the network with $g_a = 4$ performed significantly worse than the others for longer sequences. Plot **B)** shows the result of a succeeding study investigating g_a values around 12 closer. Values of $g_a = 8, 10, 12, 15$ were tried on slightly longer sequences with a higher number of symbols $|S| = 128$. 7 measurements were used to calculate averages and 95% confidence intervals using the estimated standard error for both experiments.

The persistence time of the network for different values of g_a was also estimated as a compliment to the performance metrics investigated above. This allows for comparing this aspect of the network behaviour with previous research on

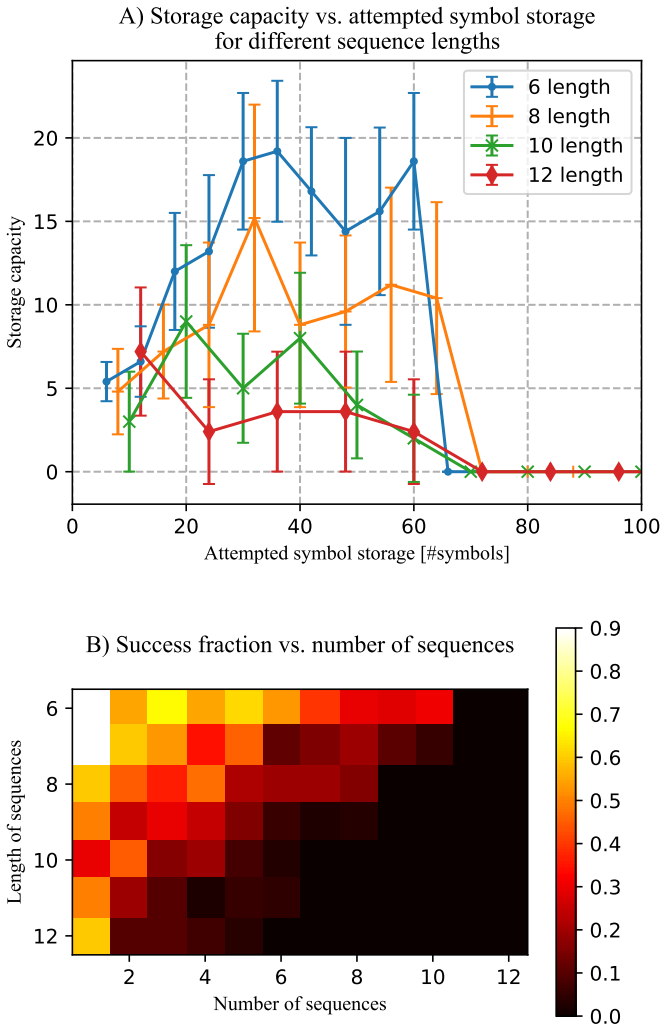


Fig. 8. Investigation of the storage capacity for a network with $H = 6$ hypercolumns and $N = 16$ units per hypercolumn. The network used an adaptation gain $g_a = 8$. Sequences was generated with $|S| = 64$ symbols from generator I. **A**) Plot of the storage capacity (defined as sequence length $\times$ sequence number $\times$ success fraction) as a function of attempted symbol storage (defined as sequence length $\times$ sequence number) for four different symbol lengths. The error bars are 95% confidence intervals using the estimated standard error based on 10 measurements. **B**) Heat plot that displays the success rate (as in percentage of sequences that were successfully recalled). A constant success fraction manifest as a linear increase in the storage capacity, while a constant or decreasing storage capacity results in a decreasing gradient to the right in the heat plot.

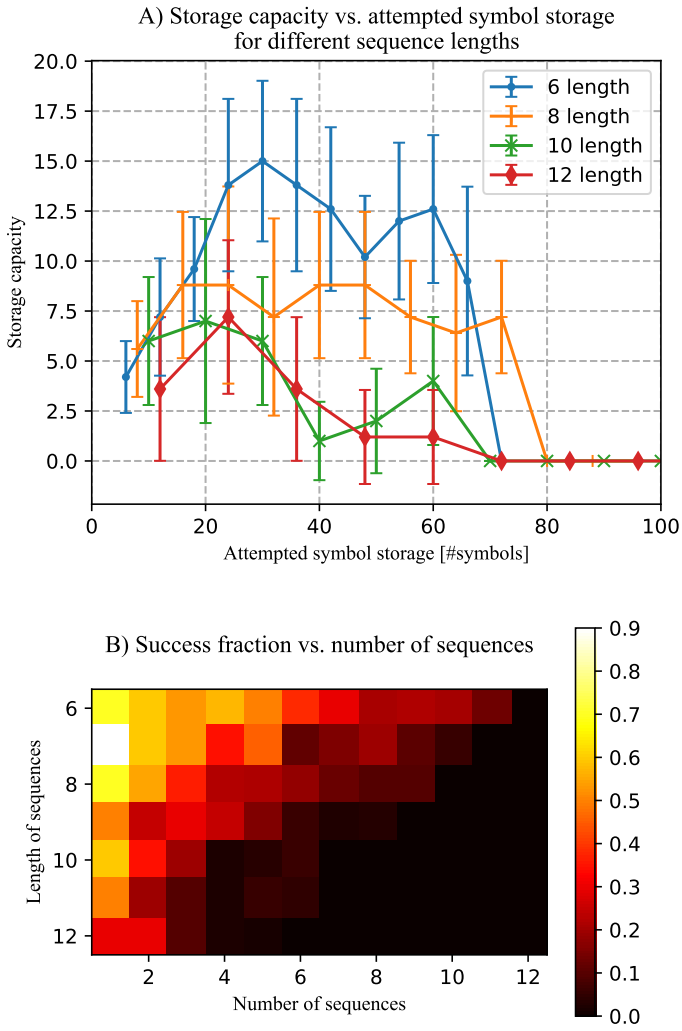


Fig. 9. Investigation of the storage capacity for a network with $H = 6$ hypercolumns and $N = 16$ units per hypercolumn. The network used an adaptation gain $g_a = 8$. Sequences was generated with $|S| = 64$ symbols from generator II, with $R_o = 33\%$. **A**) Plot of the storage capacity (defined as sequence length $\times$ sequence number $\times$ success fraction) as a function of attempted symbol storage (defined as sequence length $\times$ sequence number) for four different symbol lengths. The network performs worse on these sequences with sequential overlap than what was seen without in figure 8. The peak capacity is also less apparent as storage capacity appears to decrease with higher attempted symbol storage. The error bars are 95% confidence intervals using the estimated standard error based on 10 measurements. **B**) Heat plot that displays the success rate (as in percentage of sequences that were successfully recalled)

this network model. Experiments were run in which a network ($H = 6, N = 32$) was tasked with memorizing and recalling three sequences of length ten (drawn from sequence family II with $R_o = 33\%$). The average persistence time $\bar{T}_{\text{per}}$ was measured based on all symbol recalls excluding the first. This was done five times for each value of g_a that was investigated, and an average of the averages was calculated. This was done for several values of the adaptation gain g_a. The results are presented in figure 6. The success of the network was not considered in this measurement.

Taking this into account, as well as the study on the effect of g_a on the storage capacity of the network described earlier and shown in figure 7, $g_a = 8$ was selected as a value that combined good performance with a relatively long persistence time.

B. Learning Capacity Dependence on Sequence Distribution

After this the sequence distributions' effect on the storage capacity was investigated. Experiments were conducted in order to investigate how the sequence recall success fraction depends on the sequence length L_s and number N_s of sequences for a given network. The network structure and symbol set size were kept constant at $H = 6, N = 16$, and $|S| = 128$, while the network was tasked with storing different sets of sequences with varying size, length, and distribution.

Figure 8 shows the results for sequences pulled from family I. The network can store more symbols when memorizing shorter sequences. With only spatial overlap, the storage capacity appears to grow linearly with the number of symbols the network is trained on for a short time before plateauing at a level which could be interpreted as a peak storage capacity under the given conditions. The network generally performed better in terms of storage capacity for shorter sequences.

Figure 9 shows the results for sequences pulled from family II which introduces sequential overlap with overlap ratio $R_o = 33\%$. This study shows a decreased ability to store symbols over all tested sequence lengths as compared to the previous experiment without sequential overlap. The maximum storage capacity for a given sequence length appears to plateau at a similar attempted symbol storage as for the previous case (figure 8), but at a lower storage capacity. Performance was very low for longer sequences which is likely due to the small network size ($N = 16$, $H = 6$) used in this particular experiment.

C. Learning Capacity Dependence on Hypercolumn Size

With a good understanding of the other aspects of the network, the size of the network could finally be investigated. The network size is completely determined by two parameters: the size of the hypercolumns N, and the number of hypercolumns H. In order to investigate how the storage capacity depends on the size of the hypercolumns, the hypercolumn size N was varied while the rest of the hyperparameters were kept constant. Experiments were done with sequences drawn from a few different distributions, changing the number of sequences as well the lengths of the sequences. The results of these experiments are shown in figure 10. It is evident that the success fraction increased with increasing number of units per hypercolumn N, for all sequence lengths L_s and numbers of sequences tested.

For the shorter sequences, the network performs well starting from $N = 32$ units per hypercolumn. At this point the network is able to learn sequences equally well no matter how many sequences it is trained on with a success fraction around 75%. It is somewhat unexpected that this happens at a success fraction significantly lower than 100%. While it might just be a probabilistic coincidence since the behavior is not statistically significant, it might also mean that certain sequences are harder than others to learn for this particular network configuration. If this is the dominating cause, they would appear with around a 25% probability. The behaviour does not persist when N is increased further.

Considering figure 10 B, it is seen that the network performed poorly when trained on longer sequences and $N < 32$. At $N = 32$ units the network improves significantly but stays unreliable for more than 1-2 sequences. The success fraction also display a more varied distribution where smaller numbers of sequences have a significantly higher success fraction than larger numbers of sequences. In contrast to the behavior seen for the shorter sequences, this behavior is more expected if one assumes that the network has some storage capacity after which it fails to remember further sequences. As N is increased further performance improves, but at $N = 64$ there is still room for improvement as the success fraction remains

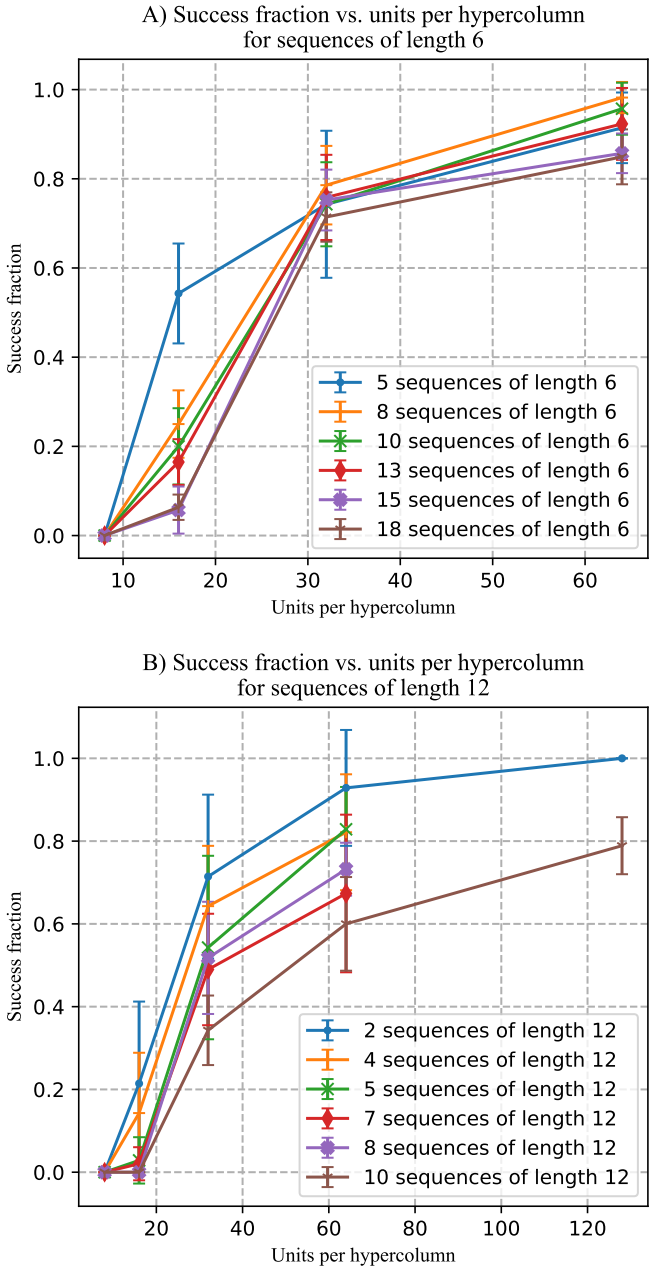


Fig. 10. The results of an experiment investigating how the storage capacity depend on the number of units per hypercolumn. Hypercolumn sizes $N = 8, 16, 32$, and 64 were tried for a network with $g_a = 8$ and $H = 6$ hypercolumns. Several number of sequences from $|S| = 128$ symbols generated from generator II with $R_o = 33\%$ were tried for each size of network. The error bars show 95% confidence intervals using the estimated standard error on averages estimated from 7 measurements. In both cases the performance of the network increases strongly with increasing hypercolumn size. **A**) For shorter sequences the network fails completely for 8 units and performs unreliably for 16. Interestingly the success fraction appear close to 75% independent on how many sequences are used for $N = 32$. This may hint that there are certain sequences that are particularly difficult to store in this configuration. At $N = 64$ the network performs well and reliably. It can successfully store all sequences that could be generated with the given symbols. **B**) For longer sequences the spread in success is more apparent. The network fails for 8 and 16 units, and performs unreliable for more than 1-2 sequences with 32 units. At 64 units the network performs better, but there is still room for improvement.

low when trying to store higher numbers of sequences. In the two final cases with $N = 128$, performance becomes much better even in the difficult case with 10 sequences. At this point however, $|S| = N$ and spatial overlap is eliminated. This thus presents a slightly different and presumably easier challenge for the network.

D. Learning Capacity Dependence on Number of Hypercolumns

Finally, the effect of the number of hypercolumns on storage capacity was investigated. H was varied, while the number of units per hypercolumn as adaptation gain were kept constant at $N = 32$ and $g_a = 8$. $|S| = 128$ symbols were used when drawing the sequences which were generated using sequence generator II with $R_o = 33\%$. The result is shown in figure 11.

In most cases, the storage capacity increases up to a point in H and then remains constant around that level. This can be interpreted as the particular network configuration reaching a peak performance level where further hypercolumns does not improve performance. Interestingly this level varies significantly between the different sequence distributions. The total number of symbols does not appear to be an important factor in this case. For example the cases with 10 sequences of length 6 and 5 sequences of length 12 both try to store the same number of symbols, but peak storage capacity is reached around 50 symbols in the first case, and a bit above 30 in the second case.

IV. Discussion and Conclusions

A. Effect of Length and Number of Sequences on Sequential Recall performance

From figures 8 and 9 it is evident that there is no simple dependence on of the success fraction on the total attempted symbol storage of a given network configuration. Instead, it depends on both the sequence length and number of sequences in a complicated way. What is clear, however, is that success fraction decreases with both increasing sequence length and increasing sequence number for a given setup.

The time spent in each epoch during training increases with the number of symbols the network is tasked with learning. The memory of sequences early in the progression is thus generally weaker than for those that come later. During the parameter exploration it was observed that this effect was most significant during for example disambiguation where the network would prefer memories closer in time over earlier memories. This effect likely played an important role during all experiments by limiting the networks capacity to handle both spatial and sequential overlap. It can however not be the only factor since if a sequence being remembered primarily depended on when it occurred in training, it would be expected that the storage capacity would be mostly independent of the sequence length. This is not at all what was seen in any of the experiments.

A complimentary effect may be that some sequences are harder to store than others. If a particular fraction of sequences generated by the sequence generator could never be stored

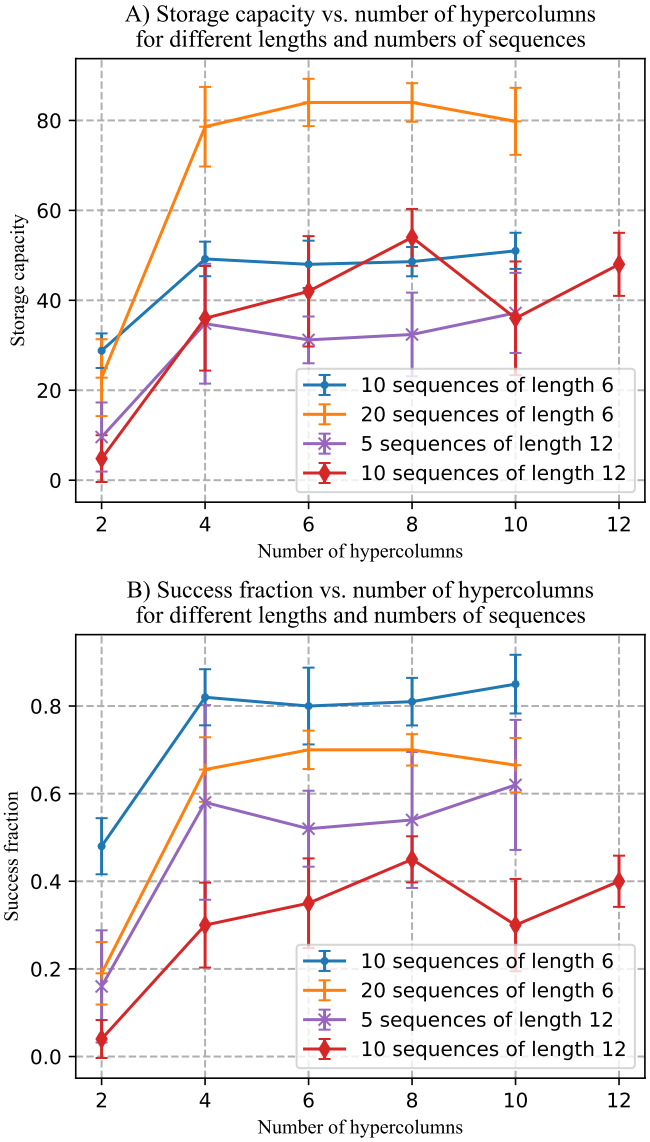


Fig. 11. The results of an experiment investigating how the storage capacity depends on the number of hypercolumns. Networks with $N = 32$ units per hypercolumn, $g_a = 8$, and a varying number of hypercolumns were tasked with memorizing a varying number of sequences with varying lengths L_s generated by sequence generator II with $R_o = 33\%$ from and $|S| = 128$ symbols. The error bars show 95% confidence intervals of an estimate of the mean using the standard error estimate calculated from 10 measurements. **A)** Initially the storage capacity quickly increases with the number of hypercolumns until reaching a steady level. This level is different for different sequence distributions. This is true for all combinations except for the one that at a first glance could be considered most difficult with 10 sequences of length 12. Interestingly for this case, the storage capacity keeps increasing steadily through all networks tried here. **B)** This plot shows the success fraction from the same experiment. While similar to plot A, it highlights the difference in behavior as compared to figure 10. While increasing the number of units per hypercolumn appears to steadily improve the network, increasing the number of hypercolumns does little after a certain point.

by the network, this would show up as a bias in the storage capacity. The results of this research offer no indication of what these sequences may be, but a possibility might be spontaneous longer sequences of sequential overlap, or a particular kind of spatial overlap. It is clear that this effect does not persist when the number of units per hypercolumn is increased. Figure 10 show that peak storage capacity increases reliably as N is

increased. Increasing the number of hypercolumns does not seem to help the network as can be seen in figure 11. This is with exception to the special case with 10 sequences of length 12 which behaves interestingly. Further research of how the network size impacts the storage capacity may shed light on the underlying mechanisms governing this peculiar behavior.

B. Strength of g_a Compared to Previous Research

Previous work has shown that a value of g_a within a factor $0 - 2.5$ in units of $\mathbf{W}$ gives good network performance [8] [7]. After the g_a optimization (see figure 7) the researchers in this project arrived at $g_a = 8$, a value that lies close to $0 - 2.5$ units of the $\mathbf{W}$-matrix seen in figure 4. The persistence times measured in this work were lower than those seen by [7]. This could potentially have a large impact on the performance of the network, especially if exposed to noise.

Investigating how other parameters impact the behavior of the network, especially τ_a which was not investigated at all in this research, could potentially alleviate this. Investigating this would be a good area for future research and could give a better foundation to the findings presented in this work. It would also be of great interest to look at how these findings stand up to the introduction of noise in a similar way to the work done in [7].

C. Effects of the Number of Units per Hypercolumn

It is evident from the results in Section III-C that the success fraction increases with increasing hypercolumn size when the network is trained on a given sequence set. Note from figure 10 that for these given sequence sets the success fraction would appear to approach 1.0 if the dimension is increased further. In most cases tested in Section III-C there are more symbol representations used than can be stored in a single hypercolumn ($|S| > N$), and it would be interesting to investigate more rigorously if this increase persists when N is increased above $|S|$. Also, to investigate how increasing the hypercolumn size past $N = 32$ affects successful recall of a set of more and longer sequences, a larger symbol set S would be needed. The extent to which this could be done was limited by the increased computation time required for a larger network and higher L_s, N_s.

D. Effects of the Number of Hypercolumns

As seen in Section IV-A the storage capacity quickly increases with increasing number of hypercolumns up to a certain level. In the experiments the storage capacity for the case with the largest number of- and longest sequences ($L_s = 12, N_s = 10$) fluctuated more than the others, which might be a result of effects emerging from the pattern fitter. The number of symbols can not in itself be the driving mechanism behind this behavior since another case with the same number of symbols ($L_s = 6, N_s = 20$) plateaued similarly to the other cases. It is possible that the number of hypercolumns for which the storage capacity reaches its peak is larger for longer sequences in particular and that larger networks are particularly good at that as compared to smaller networks (at fixed hypercolumn size). This could be an interesting area for further research.

Larger symbol sets S allow for larger sets of training data. By increasing the number of symbols as well as the size of the network, it would be possible to investigate this behavior better. In this research, computational resources ended up being a big limitation that prevented studies of larger and potentially more interesting cases.

E. Optimal Hyperparameters may Depend on Size of Network

When verifying the network, it was found that when there was only one hypercolumn a relatively high value of g_a (30-40) made the network capable of learning sequences with long overlapping subsequences. However, when the network was size was increased and more hypercolumns with spatially overlapping pattern representations were introduced, the value of g_a had to be lowered to the values tested in this report to make the network function. Eventually, as explained in section III-A, we choose $g_a = 8$ for our experiments after a systematic g_a-parameter study of a network with different values of N and H typical for our experiments. An interesting area of future research may thus be to investigate how the values of the optimal hyperparameters change as the size of the network is increased by increasing the parameters N and H. It would also be interesting to look at if this has any biological significance (for example, parameter values that resemble those of biological systems that are known to work well in practice might only start working well on larger networks). Considering that the the human brain cortex, the biological network that the model described above was inspired by [7], contains a very large number of neurons [11] compared to the networks that were investigated in this work, investigations of a model with more biologically plausible parameter choices have the potential to give insight into how the great storage capacity of biological brains emerge and function.

F. Modifying Differential Equations for Probability Traces to Better Preserve Memories From Early in the Training

The differential equations (8)-(13) each contain a negative $\mathbf{p}$-term on the right hand side, which causes each element of the $\mathbf{p}$-traces to decay over time whenever it is not stimulated. In turn, this leads to memories formed early in the training to decay over time and be weaker than memories formed later in training. This is, of course, reasonable in a model of biological memory as unused synaptic connections weakening over time is also a feature of the short term plasticity of biological neural networks. [12] However, experimenting with adding a new constant parametrizing the rate of decay of probability traces, call it τ_p^{decay}, such that

$$\tau_p \frac{d\mathbf{p}_{\text{pre/post}}^{\text{fast/slow}}}{dt} = \mathbf{z}_{\text{pre/post}}^{\text{fast/slow}} - \tau_p^{\text{decay}} \mathbf{p}_{\text{pre/post}}^{\text{fast/slow}} \tag{20}$$

would allow for slower decay. This would likely lead to the network being able to learn more as the probability traces would more easily stay high through a long training phase, and could potentially be of interest for investigating the effects of longer-lasting changes in connections between units.

G. A Theoretical Limit of the Storage Capacity

One important practical aspect of scaling the network is how it relates to real life computational time and memory usage. Maybe primarily from an applications point of view it would be desirable that the network's capabilities increases faster than its resource consumption. The memory required to store a trained network scales with the number of units as $O((HN)^2)$ (limited by the weight-matrices $\mathbf{W}^{\text{fast}}$ and $\mathbf{W}^{\text{slow}}$). The computational time for both training and recall scales with the size of the network as $O((HN)^2)$ limited by matrix operations. (Computational time for training also scales linearly with the amount of training data.) This research approaches this question from an experimental point of view, but a rough theoretical analysis is interesting and adds value to the discussion. An upper limit to the number of distinct symbols that a network could potentially memorize is given by the size of the set P' of realizable states of the activity vector $\mathbf{o}$ which in turn is given by equation (21).

$$|P'| \leq N^H = 2^{Hk} \tag{21}$$

Conversely, a strict requirement on the size of a network for storing a certain number of symbols can be formulated according to equation (22) for $|S| = |P'|$ symbols.

$$H \log_2(N) = Hk \geq \log_2(|P'|) \tag{22}$$

If this limit could be realized it would appear most beneficial to increase the number of hypercolumns H rather than the number of units per hypercolumn N since number of realizable states would increase exponentially with the number of hypercolumns, but only polynomially with the number of units per hypercolumn. For example, adding a single hypercolumn increases this limit of $|P'|$ with a factor N, while this same increase would require multiplying the number of units per hypercolumn with a factor $\sqrt[H]{N}$. An exponential increase in the number of symbols that could be handled by the network would easily win over the polynomial scaling behavior and make larger networks desirable. There are of course many other aspects that needs to be taken into consideration when choosing the size of the network, for example robustness, but even so it does not seem unreasonable to imagine that this behavior could qualitatively persist with only linear (or even polynomial) overhead. As this analysis only looks at the individual symbols it is also worth mentioning that sequential overlap has the potential to somewhat counteract potential other limitations and allow even more symbols to be stored in the context of sequences.

In this research, the apparent behavior of the studied networks seem to indicate that it is more beneficial to increase the number of units per hypercolumn rather than the number of hypercolumns. This goes against what the above analysis suggests as most optimal from an information storage perspective. The scope of the experiments done in this work is however very limited and much larger networks have to be considered to say anything about the above analysis in this subsection with confidence. A more focused study on how the number of symbols that can be used increases with the size of the network would also be very valuable. It is for example possible that a much higher number of hypercolumns could allow for a much larger symbol set size. If this was the case it would possibly permit the exponential behavior described above.

It is also possible that the way symbols are represented in the network (the choice of *Pattern Fitter* as explained in section II-G of the METHOD) could play some role in how the network behaves. While the choice used in this work naively tries to make the representations as different as possible, it may be that some intermediate level is as good or better, especially for large networks. If that is the case, more symbols could be represented. This could also allow for the exponential behavior to occur.

V. CONCLUSIONS

For the networks studied in this work the networks' ability to store and recall sequences from a given distribution depends a lot on the properties of the sequences. Given representations with spatial overlap and sequences with sequential overlap, the peak storage capacity increases with increasing number of hypercolumns up to a point after which it typically plateaus and stay approximately constant for a given set of sequences. In the experiments, the network's sequential memory performance also improved with increasing number of units per hypercolumn, appearing to approach success rate of 1.0 in the cases tested.

ACKNOWLEDGEMENTS

Thanks to our supervisor Pawel Herman for guiding us through the project and providing great feedback.

Thanks also to Axel, Douglas, Erik, and Lukas for kindly letting us use their computers to run some of our more challenging simulations.

REFERENCES

[1] E. Pastalkova, V. Itskov, A. Amarasingham, and G. Buzsáki, "Internally generated cell assembly sequences in the rat hippocampus," *Science (New York, N.Y.)*, vol. 321, pp. 1322–7, 10 2008.

[2] H. Begleiter, B. Porjesz, and W. Wang, "A neurophysiologic correlate of visual short-term memory in humans," *Electroencephalography and Clinical Neurophysiology*, vol. 87, no. 1, pp. 46–53, 1993. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0013469493901735

[3] R. Rojas, *Neural Networks - A Systematic Introduction*. Berlin: Springer, 1996.

[4] B. Vogginger, R. Schüffny, A. Lansner, L. Cederström, J. Partzsch, and S. Höppner, "Reducing the computational footprint for real-time bcpnn learning," *Frontiers in Neuroscience*, vol. 9, p. 2, 2015. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnins.2015.00002

[5] A. Sandberg, J. Tegnér, and A. Lansner, "A working memory model based on fast hebbian learning," *Network: Computation in Neural Systems*, vol. 14, no. 4, pp. 789–802, 2003, pMID: 14653503. [Online]. Available: https://doi.org/10.1088/0954-898X_14_4_309

[6] R. Echeveste and C. Gros, "Two-Trace Model for Spike-Timing-Dependent Synaptic Plasticity," *Neural Computation*, vol. 27, no. 3, pp. 672–698, 03 2015. [Online]. Available: https://doi.org/10.1162/NECO_a_00707

[7] R. Martinez, O. Kviman, A. Lansner, and P. Herman, "Sequence disambiguation with synaptic traces in associative neural," in *Artificial Neural Networks and Machine Learning - ICANN 2019*, G. Goos and J. Hartmanis, Eds. Springer, Sep. 2019, pp. 793–805.

[8] R. Martinez, A. Lansner, and P. Herman, "Probabilistic associative learning suffices for learning the temporal structure of multiple sequences." *PLoS One*, vol. 150, Aug. 2019.

[9] I. Zelenin and M. Beigi, "Sequence Disambiguation in a Brain-Like Recurrent Neural Network with Local Associative Learning," in *Kandidatexjobb i elektroteknik 2020, Kungliga Tekniska högskolan*, 2020, bsc thesis in electrical engineering.

[10] K. L. Agster, N. J. Fortin, and H. Eichenbaum, "The hippocampus and disambiguation of overlapping sequences," *Journal of Neuroscience*, vol. 22, no. 13, pp. 5760–5768, 2002. [Online]. Available: https://www.jneurosci.org/content/22/13/5760

[11] K. Saladin, *Human anatomy (3rd ed.)*. New York City: McGraw-Hill Publishing Company, 2011.

[12] R. S. Zucker and W. G. Regehr, "Short-term synaptic plasticity," *Annual Review of Physiology*, vol. 64, no. 1, pp. 355–405, 2002, pMID: 11826273. [Online]. Available: https://doi.org/10.1146/annurev.physiol.64.092501.114547

CONTEXT P – PART I
ARTIFICIAL INTELLIGENCE

POPULAR DESCRIPTION

From checkers to chess, the computers impress, it washes your dress and doesn't make a mess.

Have you ever wondered if there is a way to always win at games like Chess, Tic Tac Toe or Checkers? A magic formula, or an "ultimate strategy"? YOU might not be able to find this strategy, but a computer can. By looking ahead and finding the best move a computer can find a way to win, in most cases.

Strategic thinking is a basic human trait. Whether it is finding food to survive or trying to beat your friend at a game, our brains will try to find a way to reach the objective. Not only is this an essential skill, but we can also find joy and fulfillment in thinking up strategies and letting them play out. The game of Chess has been around since the 6th century, and is still gaining popularity to this day. Despite being around for so long, our appetite for strategic thinking is so insatiable that professionals are finding new strategies.

Recently, AI's have been able to consistently beat humans at games such as Chess and Go. Deepmind's AI is making moves that leave Go experts dumbfounded but ultimately end up being critical to winning the game. Beating humans at Go has been regarded as a huge milestone in the development of AI and many experts believed this wouldn't happen for many years.

AI might seem highly advanced and something reserved for researchers and sci-fi movies. While this held true in the past, nowadays one can find an AI in even the most unassuming of everyday objects. Say for example your laundry machine. You would be excused to think that there is nothing at all intelligent about it and yet, it can sense what temperature and rotational speed it should use. It can determine the correct dose of detergent and amount of water to ensure clean clothes, but not wasted resources.

Without a doubt, AI will have profound effects on society and the future of humanity. How exactly will the AI's used in strategy games today be applied in real life? Only the future will tell.

SUMMARY OF PROJECT RESULTS

Strategic thinking in games is a subject in which humans have been superior to computers for most of history. However in 1997 that changed, when the computer Deep-Blue beat the reigning world champion Garry Kasparov at chess, and once again in 2016, when the AI AlphaGo bested the 18 time world champion Lee Sedol in the game of Go. In 2017 an AI called Libratus decisively beat four poker pros in two player Texas hold'em over 120000 hands. This is a breakthrough since unlike Go and Chess, not all information is known. Today, AI is continuously improved upon and new research uncovers new methods for AI development.

In this context we focus on how the AI develops strategies and methods for games and how these strategies and methods can be used to model real-world scenarios.

The project groups in P1 have explored the use of a multi-agent extension of the Knowledge-Based Subset Construction (KBSC) to find strategies in multi-agent games of imperfect information against nature. Games of imperfect information are games where the agents don't have full knowledge about the current gamestate. The Knowledge-Based Subset Construction is an academic tool for transforming a single player game of imperfect information to one with perfect information in a strategy preserving way.

The project group P1A has formulated a mathematical model for representing the KBSC on Pursuit-Evasion type games on grids. A Pursuit-Evasion type game is a game with two different players: pursuers and evaders. The objective of the pursuers is to capture the evaders and the evaders's goal is to escape the pursuers. The aim of the group is to implement this mathematical model to simulate and analyze the performance of knowledge based strategies for this class of games.

The project group P1B has created an algorithm for synthesis of strategies for multi-agent games of imperfect information against nature. The algorithm has been compared to existing algorithms with respect to its speed when synthesising and testing strategies.

Future work in the same field as P1A and P1B could be to investigate the application of devised strategies and methods in real world scenarios. One aspect to investigate is how to apply these strategies in real-world scenarios. Another approach would be to study for *which* real-world scenarios the strategies could be applied to.

The project groups in P2 have created AI players for games of imperfect information. The focus has been on scaled versions of the strategy board game Stratego. This is achieved using the learning algorithm family known as counterfactual regret minimization (CFR). A computer bot repeatedly plays against itself while minimizing regrets and thereafter calculates future strategies based on that. CFR algorithms are proven to converge to theoretically optimal play, a so-called Nash equilibrium. However, in practice, this convergence is quite slow, therefore, different heuristic methods for increasing the learning rate were also tested.

The results have shown that the CFR algorithms scale poorly with game complexity and are therefore not well suited for large games.

In future work, further optimization of Stratego with other methods, such as deep neural networks, could be used to get better results. It is also of interest for military research to construct competent AI's for other imperfect information strategy games.

The work performed by the different project groups has continued upon and explored new methods and algorithms when dealing with games of imperfect information. This work broadens the range for which real-world scenarios can be modeled by games. The results for project group P2 showed that applications of memory based algorithms do not scale well with increasing complexity, at least not when they are used exclusively to solve a problem.

IMPACT ON SOCIETY AND ENVIRONMENT

The rapid development of artificial intelligence has shown that it is an extremely powerful tool, capable of solving a vast amount of problems previously thought to require human expertise. While it's too early to tell currently, many researchers in the field believe that AI will either be humanity's greatest invention, starting a new era of technological revolution and prosperity, or our worst, creating mass-unemployment, extreme societal injustice or even total collapse of civilization. The impact AI will have on society is determined, as with any tool, by the hands wielding it. It is therefore critical for AI to be developed safely and responsibly and that both researchers and users put focus towards accountability, transparency and fairness.

The **responsibility** of actions performed by an AI is difficult to pinpoint. Imagine an autonomous car that turns towards a wall to avoid a collision with a pedestrian which results in the drivers death. Is the driver at fault for his own death for not paying attention and not preventing the crash long before the AI needed to take an action? Is the AI at fault for the drivers death for making the decision to turn towards a wall, or is it the person who designed the AI who is responsible? Why did the AI in the first place value the pedestrians' life over the drivers, and who is responsible for the AI making that decision? These questions have long been difficult to answer and there will probably never be a correct answer to any of the above questions.

Context P1 and P2 revolves around games and strategies in games which can be used to model some of these real world scenarios where the responsibility of actions are difficult to determine. For example, the results from the pursuit-evasion games can be used in collision avoidance systems in automated vehicles, where above dilemmas are present. Results from

this context may also be useful in warfare scenarios, which can greatly impact human lives. In all fields where an AI is used, a difficult question will arise; who is responsible for potential undesirable decisions made by automated soldiers, doctors, drivers, etc.? To tackle these problems a balance of liability between the creator and the user has to be found.

The **consequences** of the work done in Context P lies in the possible applications. Since the project groups in this context work with finding strategies, one application (as stated above) that comes to mind is **warfare.** This application has a potentially large impact on society and individuals. Since warfare applications range from defensive to offensive uses one can argue that these applications can have both a *negative* impact*,* if the strategies/research is used to more effectively **take** people's lives, and a *positive* impact if the aim is to **save** lives. This creates a bit of a dilemma. Do the "good" applications outweigh the bad or should we treat the malicious applications as being more severe? One example of a "double-edged" utilization is an AI that uses the strategies devised to search and find a target. This AI could potentially be used to find, and save, missing people in distress but it could also be used to find and kill people. This puts light on the fact that even though the intent of the development of a product could be pure, it could still be used for evil.

Specifically for AI's in games of imperfect information, one consequence is that if the algorithm used to beat humans in poker becomes available it would have a serious negative impact on the poker community. A human poker player would statistically lose when playing against an optimized AI and considering the not so negligible sums of money involved this could spell disaster for many players' private economy.

The direct **environmental** consequences of developing an AI are negligible; the training of an AI may require some extra power but not enough to make a big difference when compared to bigger contributors such as factory production and transport. On the other hand, the indirect consequences are a lot bigger. As an example, autonomous vehicles will likely be designed to optimize efficiency, and therefore use less energy per mileage than human drivers. A big portion of the emissions generated by cars comes from stopping to a halt and starting again due to car queues and necessary things needed to regulate human drivers such as red lights. With autonomous vehicles that can communicate with the surroundings, traffic will flow smoother and necessary things to regulate human behaviours will not be needed, and emissions will reduce.

As the research, development and adoption of AI in society progresses, it is important that aspects such as **human rights, personal freedom and integrity** are not compromised. Artificial Intelligence that discriminates against a certain gender or ethnicity as a result of training data or through its application, is not ethical. Nor is it ethical for an AI to collect personal data in someone's private residence without owner's consent. At the same time, the impartiality of an AI developed with proper training data is a potential source for making impartial decisions that could help support human rights. The use of AI could reduce human involvement in scenarios where human involvement is a source of breach against personal freedom and integrity, for example when law enforcement officers search through private documents.

Knowledge Based Strategies in Grid-Based Pursuit-Evasion Games of Imperfect Information

Samuel Söderberg and Tobias Gabi Goobar

***Abstract*—Strategies in games have since long been of interest to humans, mainly to beat our friends in games such as Chess or Monopoly, but also to model real world scenarios. These strategies are often difficult to find, even more so if the players lack important information about the current state of the game. Pursuit-Evasion games are a type of games that can be used to model police chasing criminals, autonomous car collision avoidance systems and many other scenarios. It is therefore of interest to find *effective* strategies in these scenarios.**

This bachelor thesis project examined Pursuit-Evasion games of imperfect information on grids where a number of pursuers work together to capture a number of evaders whose locations are unknown. A set of knowledge-based strategies, one of them inspired by the Knowledge Based Subset Construction, were explored and analyzed. The strategies were compared against each other and against both an optimal strategy where the pursuers always were aware of the evaders whereabouts and a reference strategy where the pursuers moved randomly.

The constructed strategies proved to be efficient in comparison to the reference and in cases even close to the optimal strategy in efficiency.

***Sammanfattning*—Strategier i spel har sedan länge varit av intresse för oss människor, framförallt för att vinna mot våra kompisar i spel som Schack eller Monopol, men också för att modellera verkliga scenarion. Dessa strategier är ofta svåra att lista ut, och ännu svårare då spelarna saknar viktig information om spelets nuvarande läge. Pursuit-Evasion är en klass av spel som kan användas för att modellera polisjakter eller kollisionsundvikande system i autonoma bilar för att nämna några. Det ligger därför i vårt intresse att finna *effektiva* strategier i dessa scenarion.**

Detta kandidatsexamensarbete studerade Pursuit-Evasion spel av imperfekt information på rutnät där ett antal så kallade pursuers samarbetade för att fånga ett antal så kallade evaders vars postioner var okända. En kunskaps-representation utformades och en mängd kunskaps-baserade strategier, en inspirerad av metoden Knowledge Based Subset Construction, utforskades och testades. De olika strategierna jämfördes mot varandra och mot både en optimal strategi då pursuers hade all kunskap om var alla evaders befann sig och en referensstrategi då pursuers rörde sig slumpmässigt.

De utformade strategierna visade sig vara effektiva i jämförelse med referensen och i vissa fall till och med nära den optimala strategin i effektivitet.

***Index Terms*—Pursuit-Evasion, Knowledge-Based Subset Construction, Strategies in games, Knowledge based strategies, Game-Theory**

Supervisors: Dilian Gurov

TRITA number: TRITA-EECS-EX-2021:193

I. Introduction

When thinking of what move to make in a game of Chess or Go, we all use a different set of methods. Some try to find a set of moves that will be surely winning, regardless of the opponents actions. Others focuses completely on the opponents actions, trying to counter whatever he or she does, and when the opportunity arises sweep in and take the win. Regardless of strategy we all use some form of knowledge when choosing our next move, knowledge of the game and a sequence of moves that always wins, or knowledge of the opponents behaviours and how he or she thinks in different situations. The knowledge needed to win in a game can be anything, but for a decision to be rationalized we need some form of representation of said knowledge and a clear objective.

Inspired by the Multi-agent Knowledge-Based Subset Construction described by Gurov et al [1] and the work on Pursuit-Evasion games by Huang et al [2], the purpose of this project was to implement, simulate and analyze knowledge based strategies for Pursuit-Evasion games of imperfect information played on grids. These knowledge based strategies used a knowledge representation to make rationalized moves towards their given objective.

II. Pursuit-Evasion games

A **Pursuit-Evasion** type game is a game with two roles: *pursuers* and *evaders*. The pursuers aim to find and capture the evaders and the evaders want to avoid being captured [2]. As expected, there are a lot of different variants of Pursuit-Evasion type games and we will therefore list a set of rules and definitions to narrow down the type of game that will be discussed here. Furthermore, we will use these definitions to create a mathematical framework to model the game.

A. Preliminary Definitions

- The **arena** $\mathcal{A}$ is a finite $n \times n$ matrix.
- **Agent** is an umbrella term referring to both pursuers and evaders.
- A **location** (x, y) is a position on the 2 dimensional game-arena. Since the arena represented by a $n \times n$ matrix this corresponds to the element at row x and column y in the matrix.
- $\mathcal{L}$ is the set of all locations on the arena. Formally this is defined as

$$\mathcal{L} = \{(i, j) \mid 0 \leq i \leq n-1, 0 \leq j \leq n-1\}$$

- $P = \{p_i\}_{i=1}^{k}$ is the set of pursuers
- $E = \{e_i\}_{i=1}^{m}$ is the dynamic set of *currently remaining* evaders.

- I_e is the dynamic index set of the *currently remaining* evaders, defined as
$$I_e = \{i \mid e_i \in E\}$$
- l_{p_i} is the current location of pursuer p_i
- l_{e_i} is the current location of evader e_i
- $M(x, y)$ is the set of all **legal moves** available from location (x, y). A legal move is a move that moves the agent to a location which is either directly above, below, to the right of, or to the left of the agent's current location, as long the location is within the bounds of the arena. Formally this is defined as
$$M(x, y) = \{(x, y) + m \mid m \in \{(1, 0), (-1, 0), (0, 1), (0, 1)\} \wedge (x, y) + m \in \mathcal{L}\}$$
- $M_{e_i} = M(l_{e_i})$ is the set of all **legal moves** of evader e_i from the current location.
- $M_{p_i} = M(l_{p_i})$ is the set of all **legal moves** of pursuer p_i from the current location.
- **Making a move** means transitioning from the current location (x, y) to a location in $M(x, y)$.
- A **turn** of the game indicates all agents of a certain role making exactly one move.
- The **distance** d between two locations (i, j) and (k, l) is defined using the standard Euclidean metric on $\mathbb{R}^2$:
$$d = ||(i, k) - (k, l)|| = \sqrt{(i - k)^2 + (k - l)^2}$$

B. Rules

We will now state a list of rules that, together with the previous definition, gives the reader all information needed about how the game is played.

1) The game is played on an **arena**.
2) Each agent is given an **initial location** at the start of a game.
3) The agents take turns making moves, starting with the pursuers.
4) An evader is considered **caught** when a pursuer moves to the same location as the evader, or alternatively, that the evader moves to the same location as a pursuer. In other words, evader $e_j \in E$ is caught if
$$l_{p_i} = l_{e_j} \text{ for some } p_i \in P$$
5) After an evader gets caught it is removed from the game. This means that if evader e_j is caught e_j gets removed from E and j gets removed from I_e.
6) The game is **won** when all evaders are caught. This corresponds to when
$$E = \varnothing$$
7) Two or more pursuers can share the same location.
8) Two or more evaders can share the same location.

C. Basic example

Below follows a basic example of how a set of turns can be played with one pursuer and one evader. Assume the following

- 3×3 arena
- Initial location for pursuer $l_p = (0, 2)$
- Initial location for evader $l_e = (2, 1)$

Turn 0 - Initial locations:

$$\begin{bmatrix} 0 & 0 & p \\ 0 & 0 & 0 \\ 0 & e & 0 \end{bmatrix}$$

Turn 1 - Pursuer moves: The pursuer moves downwards.

$$\begin{bmatrix} 0 & 0 & \downarrow \\ 0 & 0 & p \\ 0 & e & 0 \end{bmatrix}$$

Turn 2 - Evader moves: The evader moves upwards.

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & e & p \\ 0 & \uparrow & 0 \end{bmatrix}$$

Turn 3 - Pursuer moves: The pursuer moves leftwards, catches the evader and wins the game.

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & p & \leftarrow \\ 0 & 0 & 0 \end{bmatrix}$$

III. Games of imperfect information

To understand the concept of imperfect information we will start by describing the concept of *perfect information*. Given a game, we say that the player has perfect information if all aspects of the game are known to the player. The player has full access to all information about the current game state. There is, in other words, no element of uncertainty for the player. For the case of the Pursuit-Evasion game described in this report a pursuer having perfect information would mean that the pursuer is at all times aware of the positions of all evaders, how many evaders are left, the layout of the arena and so on. Conversely, a game of *imperfect information* is a game where there are uncertainties about the game state for the player. In practice this means that certain game states are indistinguishable from each other for the player [1] [3]. Based on this, a few types of imperfect information in the previously defined Pursuit-Evasion game could be devised as one of the following.

- **Limited visibility for pursuers**. This means that the pursuers are not aware what all locations $l \in \mathcal{L}$ contains, i.e. if it contains an evader, a pursuer or is empty, but can observe a subset $A \subset \mathcal{L}$ of locations at all times.
- **No, or limited, knowledge of evaders moves**. The pursuers are at all times unaware of what moves the evaders will make.
- **No knowledge of amount of remaining evaders**. The pursuers are at all times unaware of how many evaders are still left on the field. An exception is that the pursers might still be able to know if the game is won (when there are *no* evaders left).
- **No knowledge of arena dimensions**. The pursuers are at all times unaware the dimensions of the arena.

Note that this is just a few examples of types of imperfect information. Below follows a description of the types of imperfect information the game examined in this report will have.

1) Visibility: We say that a location is **observable** by pursuer p_i if the pursuer is aware of what this location contains. Let $\mathcal{O}(l) \subset \mathcal{L}$ be the set of observable locations from location l. We will explore two different definitions of this set:

1) **Corridor**
   Let $l = (i, j)$ be a location.

$$\mathcal{O}(i,j) = \{(x,y) \mid (x = i \wedge 0 \leq y \leq n-1) \vee (y = j \wedge 0 \leq x \leq n-1)\}$$

   This set corresponds to all locations $l' \in \mathcal{L}$ in the same **row** or **column** as location l.
2) **Radius**
   Let $l = (i, j)$ be a location.

$$\mathcal{O}(i,j) = \{(x,y) \mid ||(x,y) - (i,j)|| \leq r\}$$

   This set corresponds to all locations $l' \in \mathcal{L}$ within a **radius** r of location l.

2) Knowledge of evaders moves: The pursuers know that the evaders follow the same rules for which moves are legal as the pursuers themselves. However, the pursuers does not know which move each evader chooses each turn.

3) Knowledge of amount of remaining evaders: The pursuers are at all times aware of how many evaders are left (The cardinality of E).

4) Knowledge of arena: The pursuers are at all times aware of the dimensions of the arena.

IV. Knowledge-Based Subset Construction

Constructing strategies in games of imperfect information has proven to be difficult. However, using the **Knowledge Based Subset Construction (KBSC)** one can transform a game of imperfect information to one with perfect information based on the agents current knowledge. One can then use the constructed game of perfect information to find strategies in the original game [3]. In a game of imperfect information certain game states are indistinguishable from each other for the player [4]. To visualize this we will show an example from our Pursuit-Evasion game.

$$\overbrace{\begin{bmatrix} 0 & 0 & p & 0 \\ 0 & 0 & 0 & 0 \\ 0 & e & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}^{A} \quad \overbrace{\begin{bmatrix} 0 & 0 & p & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & e \\ 0 & 0 & 0 & 0 \end{bmatrix}}^{B}$$

Assume we are using the *Corridor* definition of $\mathcal{O}$. Game state A is indistinguishable from game state B for the pursuer since the pursuer can only observe what is in the same row or column as itself.

By combining the indistinguishable game states into the same state we can construct a new game of perfect information [4]. As an example, we can combine the indistinguishable game states A and B from the previous example to a new state C.

$$\overbrace{\begin{bmatrix} 0 & 0 & p & 0 \\ u & u & 0 & u \\ u & u & 0 & u \\ u & u & 0 & u \end{bmatrix}}^{C}$$

where the u indicates that the contents of these locations are unknown to the pursuer.

We will now demonstrate a transformation of our previously defined game inspired by the KBSC when using the Corridor-type visibility definition. We define a new game, referred to as *The Mad Scientist Game* with a new objective and a different ruleset. Strategies and objectives in this new game can be translated to our original game.

The Mad Scientist game

In this game, some mad scientists have accidentally created a dangerous bacteria that has reproduced and spread throughout their lab. Their objective is to to stop the bacteria from spreading and capture the original bacteria to research further. The lab is represented by an $n \times n$ grid. Here are the rules of the game:

1) At the start of the game the bacteria have spread throughout the entire lab.
2) A subset of the bacteria are the **original bacteria**, these are stronger than the offspring.
3) The bacteria and scientists take turns making moves, starting with the scientists.
4) When moving, the bacteria reproduce in all locations they can move to (legal moves are defined in the same way as before).
5) The scientists are equipped with disinfectant-sprays which they spray to every location in the same **row** or **column** as their current location. Any bacteria that is not an *original bacteria* is killed instantly by being hit by the spray. The original bacteria do not get killed by the spray, but they do get *weakened*. If all original bacteria are weakened at the same time (i.e during the same turn), they are unable to provide nutrition to their offspring. This results in the demise of all non-original bacteria.
6) An original bacteria is considered **captured** when a scientist moves to the same location as the bacteria, or alternatively, that the bacteria moves to the same location as a scientist.
7) The game is won when all the original bacteria are captured.
8) The scientist can at all turns see the entire arena, but cannot tell apart an original bacteria from its offspring unless it is sprayed.
9) Two or more bacteria can share the same location.
10) Two or more scientists can share the same location.

This game is constructed from the original game by letting the pursuers be scientists and the evaders be the original bacteria. Furthermore, all locations which contents are unknown to the

pursuers are filled with non-original bacteria. The new game is a game of **perfect information** (in regards to vision) since the scientist have full vision of the arena.

A. Basic example, Mad Scientist game

In this example we will have one scientist and one original bacteria. Assume the following

- 4×4 arena
- Initial location for scientist $l_s = (0, 2)$
- Initial location for original bacteria $l_{b_o} = (2, 1)$

Turn 0 - Initial locations:

$$\begin{bmatrix} 0 & 0 & s & 0 \\ b & b & 0 & b \\ b & b_o & 0 & b \\ b & b & 0 & b \end{bmatrix}$$

Turn 1 - Scientist moves:

$$\begin{bmatrix} 0 & 0 & \rightarrow & s \\ b & b & 0 & 0 \\ b & b_o & 0 & 0 \\ b & b & 0 & 0 \end{bmatrix}$$

Turn 2 - Bacteria moves and reproduces:

$$\begin{bmatrix} 0 & 0 & 0 & s \\ b & b_o & b & 0 \\ b & b & b & 0 \\ b & b & b & 0 \end{bmatrix}$$

Turn 3 - Scientist moves:

$$\begin{bmatrix} 0 & 0 & 0 & \downarrow \\ 0 & b_o & 0 & s \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Below is a *sketch* of the proof of equivalence of the two games. To prove that the transformation of the original game to the Mad Scientist game is correct we need to prove that the two games are equivalent.

Proof. Since all rules regarding how agents move, and the layout of the arena are identical in both games we only need to prove that winning in the transformed game results in winning in the original game as well as well as that the information available to the player remains the same in the two cases.

To win in the transformed game the scientists have to capture all the original bacteria. The bacteria are captured if they are at the same location as a scientist.

To win in the original game the pursuers have to catch all evaders. The evaders are caught if they are in the same location as a pursuer.

If we map scientists to pursuers and original bacteria to evaders the equivalence of the winning condition follows trivially.

The rest the proof follows from the definition of the $\mathcal{K}$ matrix (defined later on in the report). By constructing the $\mathcal{K}$ matrix from a given game state in the original game we get the corresponding arena of the Mad Scientist game by mapping the 1:s to bacteria and pursuers to scientists. □

Finding a good strategy in the transformed game is now very easy. At every turn, we want to eliminate as many bacteria as possible. This is done by choosing the move which moves the scientist to a location with the maximum amount of bacteria in the same row and column. This strategy can be translated into the original game. Every bacteria corresponds to a position that the original bacteria could be at, which means that removing a bacteria in the transformed game removes a location where there could be an evader in the original game. Using the strategy of removing a maximum amount of bacteria in the transformed game, translates to removing a maximum amount of locations that the evaders could be at in the original game.

V. KNOWLEDGE BASED STRATEGIES

A **knowledge based strategy** is a strategy that given a knowledge state determines what move the pursuer should make. A knowledge based strategy consists of

1) A **knowledge representation**, that stores all information about each pursuer's knowledge.
2) A **knowledge update function**, which updates the knowledge for each pursuer from one turn to the next.
3) An **action mapping**, which maps each knowledge state to a specific move for the pursuer.

A. Knowledge representation

To construct a knowledge representation we first have to discuss what aspects about the game state that are known to the pursuers. As discussed above we have to take the following into account.

- The pursuers can each observe a set of locations $\mathcal{O} \subset \mathcal{L}$.
- The pursuers are aware of the dimensions of the arena (and therefore it's layout).
- The pursuers know what moves are possible to make from any location.
- The pursuers know how many evaders that are left.

This has some implications. Being able to observe a set of locations lets the pursuers draw direct conclusions about if there are any evaders within that set of locations. Furthermore, knowing what moves are legal and the layout of the arena makes it possible for the pursuer to deduce all locations where an evader **could currently be in**. When observing an evader the pursuer can calculate all possible locations the evader could be in **after the next time it moves** and can therefore draw conclusions both about the current game state and the next. Finally, the fact that the pursuers are aware of how many evaders are left lets the pursuer know if they currently can observe **all evaders on the arena**. In that case the pursuers no longer have any uncertainty about the evaders locations. We will now use this as a base for defining the knowledge representation.

The knowledge of pursuer p_i during turn $t \in \mathbb{N}_0$ is represented by two matrices $\mathcal{K}^i(t), \mathcal{P}^i(t)$ and a set of observable locations $\mathcal{O}(l_{p_i})$ from the pursuers current location l_{p_i}

1) Definitions: Let $\mathbb{K}(p_i) \subseteq \mathrm{M}_n(\mathbb{Z}_2)$ be the set of *all possible knowledge states* of pursuer p_i, where $\mathrm{M}_n(\mathbb{Z}_2)$ the set of all $n \times n$ matrices on $\mathbb{Z}_2$. Also, let $\mathbb{O}(p_i) \subset \mathcal{P}(\mathcal{L})$ be the set of *all possible sets of observable locations* of pursuer p_i, where $\mathcal{P}(\mathcal{L})$ is the power set of $\mathcal{L}$.

$\mathcal{K}^i(t) \in \mathbb{K}(p_i)$ is a an $n \times n$ **knowledge matrix** for pursuer p_i consisting of ones and zeros, where a zero represents that **no** evaders can be in that location during the current turn t, and a one represents that the evader can be in that location during the current turn t. This matrix is formally defined as follows:

$$\text{If } l_{e_j} \in \mathcal{O}(l_{p_i}) \quad \forall j \in I_e :$$
$$\mathcal{K}^i_{xy}(t) = \begin{cases} 1, & (x,y) = l_{e_j} \\ 0, & \text{otherwise} \end{cases}$$

$$\text{If } t = 0 \text{ and } \exists j \in I_e \text{ such that } l_{e_j} \notin \mathcal{O}(l_{p_i}) :$$
$$\mathcal{K}^i_{xy}(t) = \begin{cases} 0, & (x,y) \in \mathcal{O}(l_{p_i}) \text{ and } (x,y) \neq l_{e_j}, j \in I_e \\ 1, & \text{otherwise} \end{cases}$$

$$\text{If } t \equiv 1 \pmod 2 \text{ and } \exists j \in I_e \text{ such that } l_{e_j} \notin \mathcal{O}(l_{p_i}) :$$
$$\mathcal{K}^i_{xy}(t) = \begin{cases} 0, & (x,y) \in \mathcal{O}(l_{p_i}) \text{ and } (x,y) \neq l_{e_j}, j \in I_e \\ 1, & (x,y) \in \mathcal{O}(l_{p_i}) \text{ and } (x,y) = l_{e_j}, j \in I_e \\ \mathcal{K}^i_{xy}(t-1), & \text{otherwise} \end{cases}$$

$$\text{If } t \equiv 0 \pmod 2, t \neq 0, \text{ and } \exists j \in I_e \text{ such that } l_{e_j} \notin \mathcal{O}(l_{p_i}) :$$
$$\mathcal{K}^i_{xy}(t) = \begin{cases} 0, & (x,y) \in \mathcal{O}(l_{p_i}) \text{ and } (x,y) \neq l_{e_j}, j \in I_e \\ 1, & (x,y) \in \mathcal{O}(l_{p_i}) \text{ and } (x,y) = l_{e_j}, j \in I_e \\ \mathcal{P}^i_{xy}(t-1), & \text{otherwise} \end{cases}$$

Lets discuss the intuition behind this cumbersome definition. The first case, when $l_{e_j} \in \mathcal{O}(l_{p_i}) \quad \forall j \in I_e$, corresponds to the case where the pursuer p_i sees **all** evaders currently on the field simultaneously. As defined in the rules of the game, the pursuers are at all times aware of the amount of currently remaining evaders and can therefore be certain of the positions of all evaders if the amount of evaders observed is the same as the amount left on the arena. Therefore the $\mathcal{K}$ matrix will have ones at the locations of the evaders and zeros everywhere else.

If the turn number is 0, which is the initial configuration before any agent has made a move, and not all evaders are observed the $\mathcal{K}$ matrix is defined as in case 2. For this case the matrix is filled with ones except for the observed locations of p_i that does not contain any evaders, which are put to zero.

As defined in the rules, the agents take turns making moves starting with the pursuers, therefore it will be the pursuers turn to make a move when $t \equiv 1 \pmod 2$ and the evaders turn when $t \equiv 0 \pmod 2$ (when $t \neq 0$). When it is the pursuers turn to move and the pursuer p_i does not observe all remaining evaders (after moving) we get case 3 of the definition of $\mathcal{K}$. When the pursuer moves it will observe a new set of locations. Out of these locations there can either be an evader at the location or the location is empty. If the location is empty the pursuer knows that there is no evader at that position and the matrix will therefore have a zero at that position. If there instead is an evader at that location the pursuer knows that there is an evader there and that position in the matrix will have a one. For the locations that are not observed after moving we put the matrix inputs of $\mathcal{K}$ as the same as for the turn before (before the pursuer moved). These inputs are still valid since the evader has not yet moved since calculating the $\mathcal{K}$ matrix of the turn before and therefore have the same positions.

To understand the last case of the definition one needs to know the definition of the $\mathcal{P}$ matrix. We will therefore postpone the explanation of the last case until the reader is familiar with $\mathcal{P}$.

$\mathcal{P}^i(t) \in \mathbb{K}(p_i)$ is a an $n \times n$ knowledge matrix for pursuer p_i consisting of ones and zeros, where a zero represents that the evader can **not** be in that location after the *next* time the evader makes a move, and a one represents that the evader can be in that location after the *next* time the evader makes a move. This matrix is formally defined as follows:

$$\mathcal{P}^i_{xy}(t) = \begin{cases} 1, & \text{if } \mathcal{K}^i_{x-1y}(t) = 1 \\ 1, & \text{if } \mathcal{K}^i_{x+1y}(t) = 1 \\ 1, & \text{if } \mathcal{K}^i_{xy-1}(t) = 1 \\ 1, & \text{if } \mathcal{K}^i_{xy+1}(t) = 1 \\ 0, & \text{otherwise} \end{cases}$$

The $\mathcal{P}$ matrix is defined such that if there is a one in the $\mathcal{K}$ at location (x, y) (indicating that an evader *could* currently be in that location), the $\mathcal{P}$ will have ones at each location to which an evader located in (x, y) could legally move to. All other locations are put to zero since there is no possibility that an evader could move to that location given the current configuration. For that reason, the ones in $\mathcal{P}$ represent all possible locations for the evader *after* the move has been made.

Note that the $\mathcal{P}^i(t)$ matrix only becomes relevant to compute when it is the evaders turn to move, in other words when $t \equiv 0 \pmod 2$.

As previously announced, we can now discuss the intuition regarding the last case of the definition of $\mathcal{K}$. We yield case 4 when it is the evaders turn to move (when $t \equiv 0 \pmod 2, t \neq 0$) and the pursuers do not observe all evaders (after the evaders have moved). When the evader e_j makes a move we have the following four cases:

1) e_j moves **from** a location $l \notin \mathcal{O}(l_{p_i})$ **to** another location $l' \notin \mathcal{O}(l_{p_i})$
2) e_j moves **from** a location $l \notin \mathcal{O}(l_{p_i})$ **to** a location $l' \in \mathcal{O}(l_{p_i})$
3) e_j moves **from** a location $l \in \mathcal{O}(l_{p_i})$ **to** another location $l' \in \mathcal{O}(l_{p_i})$
4) e_j moves **from** a location $l \in \mathcal{O}(l_{p_i})$ **to** a location $l' \notin \mathcal{O}(l_{p_i})$

The locations in $\mathcal{O}(l_{p_i})$ can either contain an evader or be empty. If the location contains an evader the pursuer knows that an evader is in that location and that location is set to one in $\mathcal{K}$. If the location is empty the pursuer is aware that there

are no evader in that location and that location is set to 0 in $\mathcal{K}$. For the locations that are not in $\mathcal{O}(l_{p_i})$ there are ones at locations which an evader could have moved to, given what moves are legal, and zeros at the locations where no evader could possibly have moved to. This is exactly the same as the definition of the $\mathcal{P}$ for the previous turn (before the evaders moved) and therefore all the locations not in $\mathcal{O}(l_{p_i})$ are set equal to the same locations in $\mathcal{P}(t-1)$.

Even though the definition of $\mathcal{K}$ and $\mathcal{P}$ might seem hard to work with from a mathematical perspective, the programming implementation is straightforward.

B. Knowledge update function

For each turn the matrices $\mathcal{K}^i(t)$ and $\mathcal{P}^i(t)$ are updated according to the definitions above for **each** pursuer $p_i \in P$ and $\mathcal{O}(l_{p_i})$ is updated to a new set based on the new location pursuer p_i moves to i.e. the knowledge update function is a function

$$f_u : \mathbb{K}(p_i) \times \mathbb{K}(p_i) \times \mathbb{O}(p_i) \to \mathbb{K}(p_i) \times \mathbb{K}(p_i) \times \mathbb{O}(p_i)$$
$$(\mathcal{K}^i(t), \mathcal{P}^i(t), \mathcal{O}(l_{p_i})) \mapsto (\mathcal{K}^i(t+1), \mathcal{P}^i(t+1), \mathcal{O}(l'_{p_i}))$$

that maps a knowledge matrix for pursuer p_i at turn t to a knowledge matrix for the same pursuer at turn $t+1$ and the set of observable locations $\mathcal{O}(l_{p_i})$ from location l_{p_i} of the previous turn to a new set of observable locations $\mathcal{O}(l'_{p_i})$ from a new location l'_{p_i}.

C. Action mappings/Strategies

An **action mapping/strategy** $\mathcal{S}$ is function that given the current location of a pursuer, the knowledge matrix $\mathcal{K}$ and set of currently observable locations $\mathcal{O}(l_{p_i})$ returns which of the current legal moves the pursuer should make

$$\mathcal{S} : \mathcal{L} \times \mathbb{K}(p_i) \times \mathbb{O}(p_i) \to M_{p_i}$$
$$(l_{p_i}, \mathcal{K}^i(t), \mathcal{O}(l_{p_i})) \mapsto l'_{p_i}$$

Below are the different strategies that will be examined in this report.

1) $\mathcal{S}_0$ - "Clueless pursuer": At every turn the pursuer chooses a move in M_{p_i} at random.

$$\mathcal{S}_0(l_{p_i}, \mathcal{K}^i, \mathcal{O}(l_{p_i})) = m \in M_{p_i}, \quad m \text{ chosen at random}$$

Using this strategy is essentially equivalent to the pursuer using no strategy at all. The pursuer will wander around aimlessly hoping to somehow capture the evader by shear luck. For the analysis of the strategies this strategy will be used as a reference for comparison to measure the effectiveness of the other strategies.

2) $\mathcal{S}_1$ - "Tunnel vision": If the pursuer sees the evader it chooses the move in M_{p_i} which gets it closest to the evaders location. If the pursuer does **not** see the evader it chooses a move in M_{p_i} at random. For pursuer p_i this means that if $l_{e_j} \in \mathcal{O}(l_{p_i})$ for some $j \in I_e$, let $d_j = ||l_{e_j} - l_{p_i}||$ denote the distance to l_{e_j} from l_{p_i}, we have

$$\text{If } l_{e_j} \in \mathcal{O}(l_{p_i}) \text{ for some } j \in I_e :$$
$$\mathcal{S}_1(l_{p_i}, \mathcal{K}^i, \mathcal{O}(l_{p_i})) = l_{p_i} + \frac{1}{d_j}(l_{e_j} - l_{p_i})$$
$$\text{where } j \text{ is chosen such that } d_j \leq d_k \ , \forall k \neq j.$$

$$\text{Else if } l_{e_j} \notin \mathcal{O}(l_{p_i}) \ \forall j \in I_e$$
$$\mathcal{S}_1(l_{p_i}, \mathcal{K}^i, \mathcal{O}(l_{p_i})) = m \in M_{p_i}, \quad m \text{ chosen at random}$$

3) $\mathcal{S}_2$ - "Coordinated tunnel vision": The pursuer moves exclusively in either horizontal or vertical directions until it sees an evader. When the pursuer sees any evaders, it moves towards the closest evader. This strategy chooses a move the same way $\mathcal{S}_1$ chooses a move whenever the pursuer sees an evader. If the pursuer can not see any evader, the possible moves is determined by the pursuer's index, the i in p_i. If $i \equiv 0 \ (mod\ 2)$ the possible moves are limited to the horizontal directions, and if $i \equiv 1 \ (mod\ 2)$ the pursuer's moves are limited to the vertical directions.

$$\text{In } p_i$$
$$\text{If } i \equiv 0 \pmod 2 :$$
$$M_{p_i}^{\mathcal{S}_2} = \{(x,y) + m \mid m \in \{(1,0),(-1,0)\} \wedge (x,y) + m \in \mathcal{L}\}$$
$$\text{Else if } i \equiv 1 \pmod 2 :$$
$$M_{p_i}^{\mathcal{S}_2} = \{(x,y) + m \mid m \in \{(0,1),(0,-1)\} \wedge (x,y) + m \in \mathcal{L}\}$$

The move is chosen randomly from the set $M_{p_i}^{\mathcal{S}_2}$

$$\mathcal{S}_2(l_{p_i}, \mathcal{K}^i, \mathcal{O}(l_{p_i})) = m \in M_{p_i}^{\mathcal{S}_2}, \quad m \text{ chosen at random}$$

4) $\mathcal{S}_3$ - "Removing ones": The aim of this strategy is to remove as many uncertainties as possible regarding the location of the evader at every turn. The pursuer chooses the move in M_p that **removes the most amount of 1's** in the $\mathcal{K}$ matrix.

$$\mathcal{S}_3(l_{p_i}, \mathcal{K}^i, \mathcal{O}(l_{p_i})) = m \in M_{p_i} \text{ such that}$$
$$\sum_{x,y} \left(\mathcal{K}^i_{xy}(t) - \mathcal{K}^i_{xy}(t+1)\right) \text{ is maximized}$$

Note that this in fact is the strategy that was found in the transformed game, *The Mad Scientist game*.

D. Knowledge representation and strategy example

Below is an example of how a set of turns can be played and how the knowledge is represented for the case with one pursuer p and one evader e.
Assume the following:

- 4×4 arena.
- The initial location for pursuer is $l_p = (0,2)$.
- The initial location for evader is $l_e = (2,1)$.
- The pursuer is using the $\mathcal{S}_3$ strategy.
- $\mathcal{O}(l_{p_i})$ is defined using the *Corridor* definition.

Turn 0 - Initial locations: The pursuer does not see the evader and has no memory of its earlier locations. Every location except for the ones the pursuer can observe are possible locations for the evader. The $\mathcal{P}$ matrix indicates every location the evader can be in next turn, which in this case is the whole arena except the pursuer's current location.

$$\overbrace{\begin{bmatrix}0&0&p&0\\0&0&0&0\\0&0&0&0\\0&e&0&0\end{bmatrix}}^{\mathcal{A}(0)} \quad \overbrace{\begin{bmatrix}0&0&0&0\\1&1&0&1\\1&1&0&1\\1&1&0&1\end{bmatrix}}^{\mathcal{K}(0)} \quad \overbrace{\begin{bmatrix}1&1&0&1\\1&1&1&1\\1&1&1&1\\1&1&1&1\end{bmatrix}}^{\mathcal{P}(0)}$$

Turn 1: Since any of the three possible moves in M_p removes equal amount of 1:s, the move is chosen at random. After moving, the pursuer still does not see the evader. All previous 1:s in $\mathcal{K}$ remain except for the ones in the locations which are now visible to the pursuer. The $\mathcal{P}$ matrix is updated according to the definition.

$$\overbrace{\begin{bmatrix}0&0&\downarrow&0\\0&0&p&0\\0&0&0&0\\0&e&0&0\end{bmatrix}}^{\mathcal{A}(1)} \quad \overbrace{\begin{bmatrix}0&0&0&0\\0&0&0&0\\1&1&0&1\\1&1&0&1\end{bmatrix}}^{\mathcal{K}(1)} \quad \overbrace{\begin{bmatrix}0&0&0&0\\1&1&0&1\\1&1&1&1\\1&1&1&1\end{bmatrix}}^{\mathcal{P}(1)}$$

Turn 2: The evader moves into the pursuer's vision and the $\mathcal{K}$ matrix is updated to only have a 1 in the evader's current location due to it being the only evader on the arena. The $\mathcal{P}$ matrix is then updated to include all possible locations for the evader during the next turn.

$$\overbrace{\begin{bmatrix}0&0&0&0\\0&0&p&0\\0&0&0&0\\0&\rightarrow&e&0\end{bmatrix}}^{\mathcal{A}(2)} \quad \overbrace{\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&1&0\end{bmatrix}}^{\mathcal{K}(2)} \quad \overbrace{\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&1&0\\0&1&0&1\end{bmatrix}}^{\mathcal{P}(2)}$$

Turn 3: The pursuer takes a step towards the evader due to it being the only location containing a 1 in $\mathcal{K}$. The $\mathcal{K}$ matrix is not altered due to the evader still being in the pursuer's sight.

$$\overbrace{\begin{bmatrix}0&0&0&0\\0&0&\downarrow&0\\0&0&p&0\\0&0&e&0\end{bmatrix}}^{\mathcal{A}(3)} \quad \overbrace{\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&1&0\end{bmatrix}}^{\mathcal{K}(3)} \quad \overbrace{\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&1&0\\0&1&0&1\end{bmatrix}}^{\mathcal{P}(3)}$$

Turn 4: The evader moves out of the pursuer's sight and the $\mathcal{K}$ matrix is updated using the $\mathcal{P}$ matrix from the previous turn.

$$\overbrace{\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&p&0\\0&0&\rightarrow&e\end{bmatrix}}^{\mathcal{A}(4)} \quad \overbrace{\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&1&0&1\end{bmatrix}}^{\mathcal{K}(4)} \quad \overbrace{\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&1&0&1\\1&0&1&0\end{bmatrix}}^{\mathcal{P}(4)}$$

The game will go on until the pursuer catches the evader and the $\mathcal{K}$ and $\mathcal{P}$ matrices will each turn be updated according to this example.

VI. Multi-agent Expansion

We will now continue by expanding the previously defined theory to a version of the game where the pursuers can communicate and cooperate. In this multiplayer scenario multiple pursuers work together to catch all evaders, and by doing that achieving victory.

To construct our mathematical framework for the multi-agent expansion we will use the **Multi-agent Knowledge-Based Subset Construction (MKBSC)** defined by [1] as inspiration. In [1] the MKBSC is constructed using four stages: **Projection, Expansion, Composition** and **Partition**. First, we will briefly describe these concepts and then we explain how our method for expanding the game to multiple-agents is inspired by this method.

1) **Projection:** The original multiplayer game containing all agents is projected onto several singleplayer games, one for each agent.
2) **Expansion:** The projected singleplayer games are expanded using the KBSC.
3) **Composition:** The expanded individual games are combined using a product construction.
4) **Partition:** From the composited product, each agent's observations are determined.

Our first step is inspired by the projection and expansion steps. Each pursuer p_i has its **own** knowledge representation consisting of two knowledge matrices $\mathcal{K}^i, \mathcal{P}^i$ and a set of observable locations $\mathcal{O}(l_{p_i})$. Through communication, the pursuers can share their knowledge, a composition of their knowledge is created. From this new composited knowledge and the pursuers' current positions, a set of moves is partitioned to the pursuers.

A. Definitions

We will now expand our knowledge framework for interaction between multiple agents. Lets define the **communication graph** $G_c = \langle V_c, E_c \rangle$ as follows. For every pursuer $p_i \in P$ there is a corresponding vertex $v_i \in V_c$. Furthermore there is an edge between vertex v_i and v_j **if and only if** $l_{p_j} \in \mathcal{O}(l_{p_i})$. We will now define a binary relation $\sim_c$ on $R \subseteq P \times P$ as follows.

$$p_i \sim_c p_j \iff \text{there is a path}^1 \text{ in } G_c \text{ from } v_i \text{ to } v_j, \quad p_i, p_j \in P$$

This means that two pursuers are **related** if and only if they can observe each other through a chain of pursuers. This relation is an **equivalence relation** and therefore defines **equivalence classes** on P:

$$[p_i] = \{p_j \in P \mid p_i \sim_c p_j\}$$

[1]We consider a definition of a path where the vertices do **not** have to be distinct.

Proof. The relation is **reflexive** since $l_{p_i} \in \mathcal{O}(l_{p_i}) \iff$ there is an edge between vertex v_i and itself (a loop) in $G_c \iff$ there is a path in G_c from vertex v_i to itself $\forall p_i \in P$.

The relation is **symmetric** since $p_i \sim_c p_j \iff$ there is a path from vertex v_i to v_j in $G_c \iff p_j \sim_c p_i, \quad \forall p_i, p_j \in P$.

The relation is **transitive** since $p_i \sim_c p_j \wedge p_j \sim_c p_k \implies$ there is edge between v_i and v_j and an edge between v_j and v_k in $G_c \iff$ there is a path between v_i and v_k in $G_c \iff p_i \sim_c p_k, \quad \forall p_i, p_j, p_k \in P$ □

Furthermore, we will define a binary operation on $\mathbb{K}(p_i)$. Let $\otimes$ denote the Hadamard product on $\mathbb{K}(p_i)$ defined as

$$(A \otimes B)_{ij} = A_{ij} \cdot B_{ij}, \quad A, B \in \mathbb{K}(p_i)$$

Note that $\mathbb{K}(p_i)$ is closed under $\otimes$.

VII. Communication

When introducing cooperating pursuers into the game we need to define certain *Rules of Communication*. These rules dictate how and when the pursuers should share knowledge of the current game state with each other.

Note that in the case of the trivial choice of rules where the pursuers can communicate perfectly with each other at every turn (in any game state) the game actually reduces to a single player game. It would be equivalent to there being one player controlling all pursuers, and therefore the multiplayer element would be lost. For this reason we will not consider this choice of communication rules.

A. Definition of Rules of Communication

Knowledge is shared between pursuer p_i and p_j **if and only if** $p_i \sim_c p_j$.

In practice, this means that the pursuers share knowledge with all other pursuers they can observe. **Furthermore**, all pursuers receiving this knowledge can also share this with all other pursuers *they* observe. For example, lets say that we have three pursuers: p_i, p_j, p_k. Assume that p_i observes p_j and p_i does not observe p_k. If p_j observes p_k then p_i would share knowledge with p_j which would in turn share the knowledge to p_k (and vice versa). This means that all three pursuers in this case share their knowledge with each other.

B. Knowledge sharing

We define knowledge sharing between pursuer p_i and p_j as replacing $\mathcal{K}^i(t)$ and $\mathcal{K}^j(t)$ with

$$\mathcal{K}^i(t) \otimes \mathcal{K}^j(t)$$

This means that the knowledge matrices of p_i and p_j gets replaced with a matrix with ones at the positions where **both** $\mathcal{K}^i(t)$ and $\mathcal{K}^j(t)$ have ones, and zeros otherwise.

More generally, when multiple pursuers are eligible to share knowledge with each other simultaneously we define their updated shared knowledge as

$$\mathcal{K}^i(t) = \bigotimes_{p_j \in [p_k]} \mathcal{K}^j(t), \quad \forall p_i \in [p_k], \quad p_k \in P$$

where $[p_k]$ is the equivalence class of p_k in regards to $\sim_c$

C. Multiplayer example

Next is an example of how a set of turns can be played and how the knowledge is represented for a multiplayer case with two pursuers p_1, p_2 and two evaders e_1, e_2.
Assume the following:

- 4×4 arena
- The initial location for pursuer p_1 is $l_{p_1} = (0, 0)$
- The initial location for pursuer p_2 is $l_{p_2} = (1, 3)$
- The initial location for evader e_1 is $l_{e_1} = (3, 1)$
- The initial location for evader e_2 is $l_{e_2} = (3, 2)$
- All pursuers are using the $\mathcal{S}_3$ strategy.

Turn 0 - Initial locations: The pursuers do not see any evaders.

$$\overbrace{\begin{bmatrix} p_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & p_2 \\ 0 & 0 & 0 & 0 \\ 0 & e_1 & e_2 & 0 \end{bmatrix}}^{\mathcal{A}(0)} \quad \overbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}}^{\mathcal{K}^1(0)} \quad \overbrace{\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}}^{\mathcal{K}^2(0)}$$

$$\overbrace{\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}}^{\mathcal{P}^1(0)} \quad \overbrace{\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}}^{\mathcal{P}^2(0)}$$

Turn 1 - Pursuers move: The two pursuers move to eliminate as many 1:s as possible in $\mathcal{K}$. p_2 sees an evader and updates its knowledge. p_1 and p_2 can observe each other and therefore share their knowledge. The $\mathcal{K}$ matrices are updated and then the $\mathcal{P}$ matrices.

$$\overbrace{\begin{bmatrix} \downarrow & 0 & 0 & 0 \\ p_1 & 0 & p_2 & \leftarrow \\ 0 & 0 & 0 & 0 \\ 0 & e_1 & e_2 & 0 \end{bmatrix}}^{\mathcal{A}(1)} \quad \overbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}}^{\mathcal{K}^1(1)} \quad \overbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}}^{\mathcal{K}^2(1)}$$

$$\overbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}}^{\mathcal{P}^1(1)} \quad \overbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}}^{\mathcal{P}^2(1)}$$

Turn 2 - Evaders move: The evaders move randomly in any direction. The $\mathcal{K}$ matrix of all the pursuers is updated using the $\mathcal{P}$ matrix from the previous turn. Finally, $\mathcal{P}$ is updated using the new $\mathcal{K}$ matrices.

$$\overbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ p_1 & 0 & p_2 & 0 \\ 0 & e_1 & 0 & 0 \\ 0 & \uparrow & \rightarrow & e_2 \end{bmatrix}}^{\mathcal{A}(2)} \quad \overbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}}^{\mathcal{K}^1(2)} \quad \overbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}}^{\mathcal{K}^2(2)}$$

$$\overbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}}^{\mathcal{P}^1(2)} \quad \overbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}}^{\mathcal{P}^2(2)}$$

VIII. A Set-up for Experimental Evaluation

A. Implementation

The mathematical framework of the Pursuit-Evasion game was implemented, along with the knowledge based strategies, in **Python**. The implemented framework can be found at: https://gits-15.sys.kth.se/ssoderbe/KEX_P1. Object-oriented programming was used in the implementation. The Pursuit-Evasion game framework used for simulation was implemented as its own class with methods to move agents, get current positions, show a grid of the current game, and to see if the game is won. A separate class which handles the construction of the knowledge matrices $\mathcal{K}$ and $\mathcal{P}$ was also implemented. The gameplay-logic, where these objects were combined with our definitions of imperfect information, was implemented as a third class with the different strategies as outside functions.

B. Analyzing strategies

The strategies were assessed based on 3 different criteria

1) How well the strategy's performance scales with the **size of the arena**, n.
2) How well the strategy's performance scales with the **number of pursuers**, $|P|$.
3) How well the strategy's performance scales with the **number of evaders**, $|E|$.

Here we define the performance of a strategy as the *average number of turns needed for the pursuers to win.*

To determine how the strategies performed in comparison to each other the following graphs were produced for each type of imperfect information.

1) Increasing-size graph: The Increasing-size graph displays how well the strategies perform as the size of the arena increases. The average number of turns needed for the pursuers to win over a number of games is plotted against the size of the arena. The following specifications were used when generating the plots.

- **Size range:** Sizes of $n = 2, ..., 10$ where tested. Higher values of n where not investigated as the computation time grows very quickly when increasing the size.
- **Games:** The amount of turns needed to win were computed as the average of 9000 games per size n.
- **Number of pursuers:** The number of pursuers where chosen to be $|P| = 2$. This number was chosen such that the number of pursuers would not be more than half of the number of locations for the smallest size $(n = 2)$.
- **Number of evaders:** The number of evaders was chosen to be $|E| = 2$. Like in the case of the pursuers, this number was chosen such that the number of evaders would not be more than half of the number of locations for the smallest size $(n = 2)$.
- **Initial locations:** The initial locations of both the pursuers and evaders were chosen randomly from $\mathcal{L}$.
- **Evader behaviour:** The evaders used the $\mathcal{S}_0$ strategy i.e. moved randomly.

2) Increasing-Pursuers graph: The Increasing-Pursuers graph displays how quickly the number of turns needed to win decreases using the different strategies as the number of the pursuers increases. The size of the arena and the number of evaders remain constant. The average number of turns needed for the pursuers to win over a number of iterations is plotted against the number of pursuers. Below are the specifications used to create the plots.

- **Size:** The size of the arena was chosen as $n = 10$.
- **Games:** The amount of turns needed to win were computed as the average of 9000 games per value of the size n.
- **Range of number of pursuers:** Number of pursuers up to $|P| = 100$ where tested.
- **Number of evaders:** The number of evaders where chosen to be $|E| = 2$.
- **Initial locations:** The initial locations of both the pursuers and evaders were chosen randomly from $\mathcal{L}$.
- **Evader behaviour:** The evaders used the $\mathcal{S}_0$ strategy i.e. moved randomly.

3) Increasing-Evaders graph: The Increasing-Evaders graph displays how quickly the number of turns needed to win increases using the different strategies as the number of the evaders increases. The size of the arena and the number of pursuers remain constant. The average number of turns needed for the pursuers to win over a number of games is plotted against the number of evaders. Below are the specifications used to create the plots.

- **Size:** The size of the arena was chosen as $n = 10$.
- **Games:** The amount of turns needed to win were computed as the average of 9000 games per value of the size n.
- **Range of number of evaders:** Number of evaders up to $|E| = 100$ where tested.
- **Number of pursuers:** The number of pursuers was chosen to be $|P| = 2$.
- **Initial locations:** The initial locations of both the pursuers and evaders were chosen randomly from $\mathcal{L}$.
- **Evader behaviour:** The evaders used the $\mathcal{S}_0$ strategy i.e. moved randomly.

C. Lower bound on performance using optimal strategy

When analyzing and comparing the strategies an additional strategy was included. This strategy is allowed to access the

locations of the evaders at all times. The strategy chooses the move that minimizes the distance to the closest evader. This means that the strategy will always win in a minimal number of turns. This strategy can therefore be seen as an ”optimal strategy”. This strategy was included as a reference, to visualize how effective the strategies were compared to a lower bound.

IX. Result & Discussion

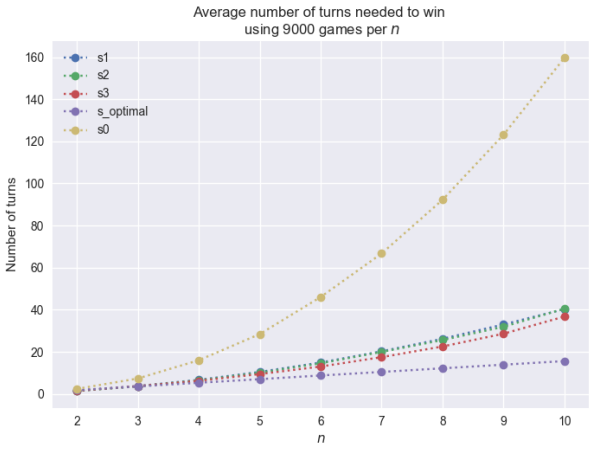


Fig. 1. Increasing-Size graph with $|P| = 2, |E| = 2$ for **Corridor**-type definition of $\mathcal{O}(l_{p_i})$.

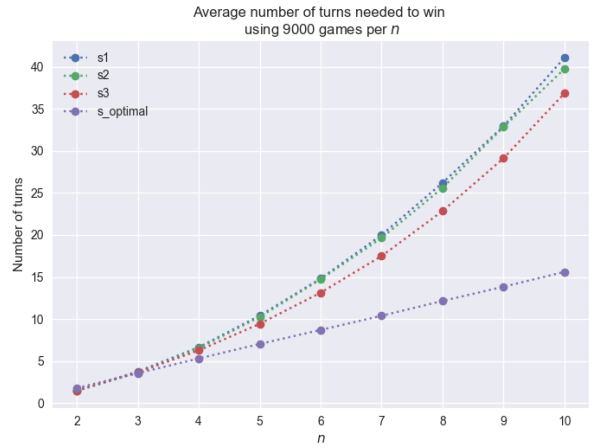


Fig. 2. Increasing-Size graph with $|P| = 2, |E| = 2$ for **Corridor**-type definition of $\mathcal{O}(l_{p_i})$ In this graph $\mathcal{S}_0$ was omitted to better display the differences between $\mathcal{S}_1, \mathcal{S}_2$ and $\mathcal{S}_3$.

In all criteria $\mathcal{S}_1, \mathcal{S}_2$ and $\mathcal{S}_3$ perform considerably better than $\mathcal{S}_0$ as can be seen in Fig. 1, 3, 5, 6, 8 and 10. This shows that the knowledge-based strategies are in fact very effective compared to using no strategy at all.

In Fig. 2. we can clearly see that when n increases while $|P|$ and $|E|$ are constant, $\mathcal{S}_3$ is more effective than $\mathcal{S}_2$ which is slightly more effective than $\mathcal{S}_1$. $\mathcal{S}_3$ works to eliminate possible locations for the evaders and by that singling out the evaders locations. The results shows that this is a better strategy than

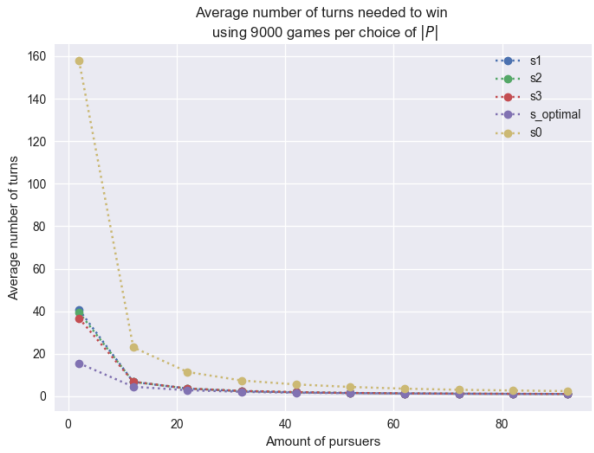


Fig. 3. Increasing-Pursers graph with $n = 10, |E| = 2$ for **Corridor**-type definition of $\mathcal{O}(l_{p_i})$.

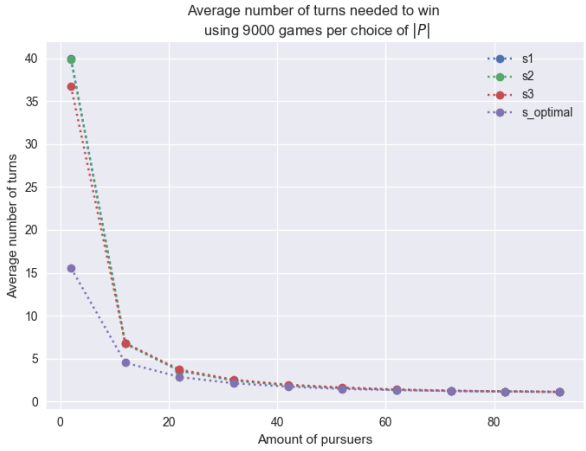


Fig. 4. Increasing-Pursers graph with $n = 10, |E| = 2$ for **Corridor**-type definition of $\mathcal{O}(l_{p_i})$. In this graph $\mathcal{S}_0$ was omitted to better display the differences between $\mathcal{S}_1, \mathcal{S}_2$ and $\mathcal{S}_3$.

wandering randomly until an evader is spotted. Many of the moves $\mathcal{S}_1$ and $\mathcal{S}_2$ chooses are moves in random directions due to the pursuers not seeing any evaders. When the size of the arena increases, the probability of a pursuer to observe an evader decreases and likewise the number of moves taken in random directions for $\mathcal{S}_1$ and $\mathcal{S}_2$.

When changing to the **Radius**-type definition of $\mathcal{O}(l_{p_i})$ we can see in Fig. 7 that $\mathcal{S}_3$ is even more effective than $\mathcal{S}_1$ and $\mathcal{S}_2$ when n increases. The radius r used in these simulations is set to $r = 0.3 \cdot n$, which when implemented is rounded down to nearest integer. This creates a behaviour where the number of turns needed to win dips when $n = 3 \cdot l + 1$ where $l \in \mathbb{N}$. For example, when $n = 4, 5, 6$ we have that $r = 1$, and when n increases to $n = 7$, r increases to $r = 2$. n will always be larger than r, so an increase of 1 in n will always be lesser in proportion than an increase of 1 in r, i.e.

$$\frac{n+1}{n} < \frac{r+1}{r}$$

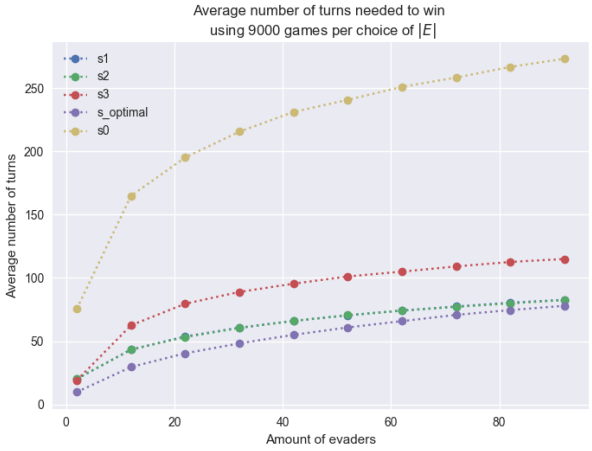


Fig. 5. Increasing-Evaders graph with $n = 10, |P| = 2$ for **Corridor**-type definition of $\mathcal{O}(l_{p_i})$.

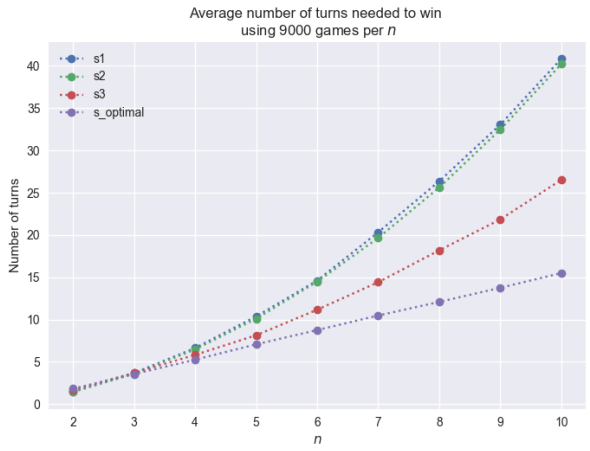


Fig. 7. Increasing-Size graph with $|P| = 2, |E| = 2$ for **Radius**-type definition of $\mathcal{O}(l_{p_i})$ with $r = 0.3 \cdot n$ In this graph $\mathcal{S}_0$ was omitted to better display the differences between $\mathcal{S}_1, \mathcal{S}_2$ and $\mathcal{S}_3$.

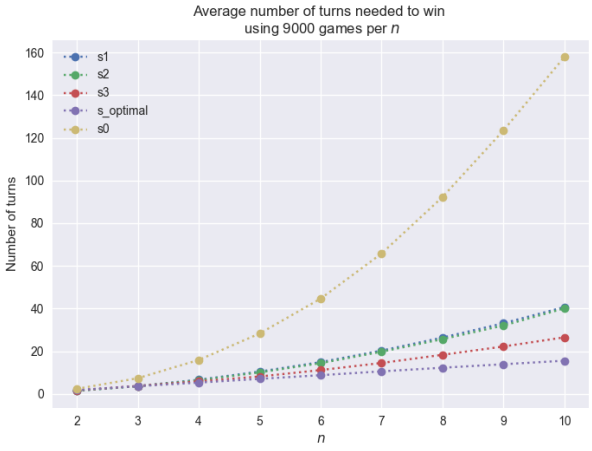


Fig. 6. Increasing-Size graph with $|P| = 2, |E| = 2$ for **Radius**-type definition of $\mathcal{O}(l_{p_i})$ with $r = 0.3 \cdot n$.

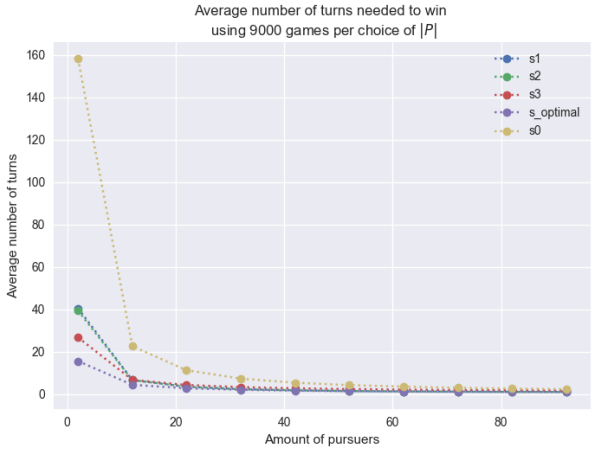


Fig. 8. Increasing-Pursers graph with $n = 10, |E| = 2$ for **Radius**-type definition of $\mathcal{O}(l_{p_i})$ with $r = 0.3 \cdot n$.

The proportionally larger increase in r increases the efficiency of the strategy. An increase in radius also increases the possibility of pursuers observing each other, which amplifies the increase in efficiency for $\mathcal{S}_3$.

From Fig. 4 and Fig. 9 we can observe that $\mathcal{S}_1, \mathcal{S}_2$ and $\mathcal{S}_3$ have very similar performance when the number of pursuers increases for both the **Corridor**- and **Radius**-type definitions of $\mathcal{O}(l_{p_i})$. This can be explained by the fact that all three of these strategies uses the same logic when choosing a move in the case when all evaders currently on the arena is seen simultaneously. Due to the number of pursuers being large, the probability of observing an evader increases. Furthermore, the probability of sharing knowledge also increases. A consequence of the number of pursuers becoming very large compared to the amount of locations on the arena is that each evader is almost always observed by at least one pursuer. This ensures that some pursuers always chooses the move which gets them closest to the nearest evader. This leads to these strategies having a performance very close to the optimal strategy for high values of $|P|$.

When increasing the number of evaders $|E|$ in Fig. 5 and Fig. 10 while keeping the arena size n and the number of pursuers $|P|$ constant, we see that $\mathcal{S}_3$ takes quite a lot more turns to win than $\mathcal{S}_1$ and $\mathcal{S}_2$. We can also see that $\mathcal{S}_1$ and $\mathcal{S}_2$ converges to the optimal solution for larger values of $|E|$. This is due to the fact that when the amount of evaders tends to $|E| = 100$ the entire arena is almost filled with evaders. This ensures that the pursuers consistently can observe at least one evader. In the case of $\mathcal{S}_1$ and $\mathcal{S}_2$ the pursuer then chooses to move towards the closest visible evader. The difference between the optimal strategy and these two therefore become very small, since the optimal solution chooses its moves the same way as $\mathcal{S}_1$ and $\mathcal{S}_2$ would do if the pursuers could observe all evaders at all turns. However, when $\mathcal{S}_3$ uses its knowledge matrix to determine its next move, it doesn't distinguish between an evader being in a location and the *possibility* of an evader being in a location, it focuses solely on removing as many ones as possible in the $\mathcal{K}$ matrix. This may result in $\mathcal{S}_3$ choosing a move which removes

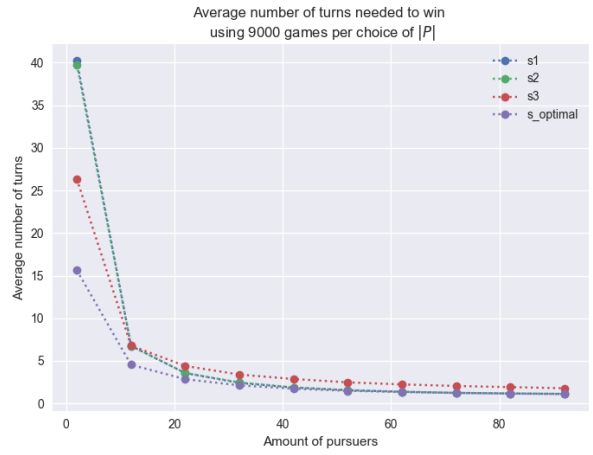


Fig. 9. Increasing-Pursers graph with $n = 10, |E| = 2$ for **Radius**-type definition of $\mathcal{O}(l_{p_i})$ with $r = 0.3 \cdot n$. In this graph $\mathcal{S}_0$ was omitted to better display the differences between $\mathcal{S}_1, \mathcal{S}_2$ and $\mathcal{S}_3$.

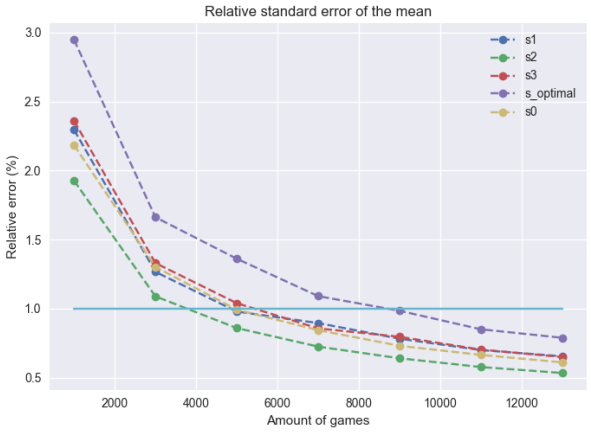


Fig. 11. RSEM graph with $|P| = 2, |E| = 2, n = 2$ for **Corridor**-type definition of $\mathcal{O}(l_{p_i})$.

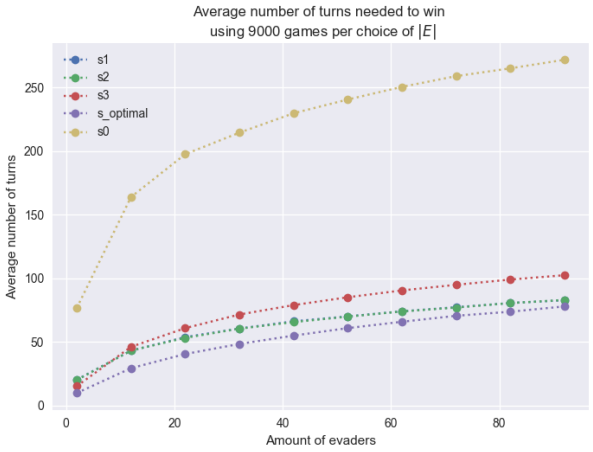


Fig. 10. Increasing-Evaders graph with $n = 10, |P| = 2$ for **Radius**-type definition of $\mathcal{O}(l_{p_i})$ with $r = 0.3 \cdot n$.

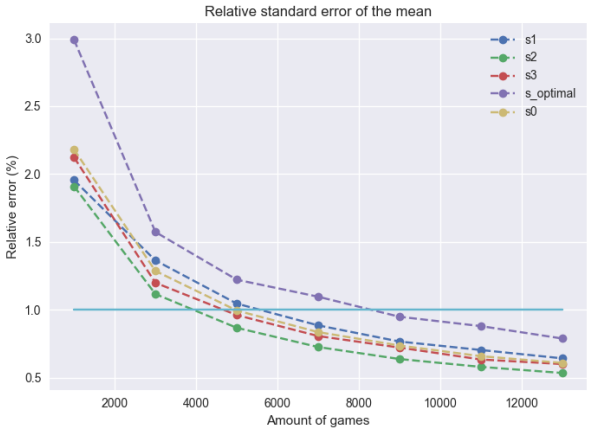


Fig. 12. RSEM graph with $|P| = 2, |E| = 2, n = 2$ for **Radius**-type definition of $\mathcal{O}(l_{p_i})$.

a lot of ones instead of a move that would relocate the pursuer to the same location as an evader.

A. Statistical accuracy

Because of the random initial locations of the agents and the non-determinism of both the evaders' movements and the pursuers' strategies the performance can vary a lot for different games. To measure the statistical accuracy of the findings the Relative Standard Error of the Mean (RSEM) was used. To ensure reliability of the results the amount of games that where averaged over when computing a strategy's performance was chosen to be high enough for the RSEM to be below 1%.

Fig. 11 and Fig. 12 shows how the RSEM decreases as the number of games averaged over increases. As one can see from the plot the RSEM falls below 1% at around 9000 games.

As RSEM is not invariant in regards to n Fig. 13 and Fig. 14 were generated to show that the choice of 9000 games ensures that RSEM is below 1% for all sizes tested in the analysis.

B. Real world application

Imagine a scenario where a search party is trying to find a missing person, possibly during night when vision is limited. Furthermore, assume that the terrain is known. The search party might have some way of looking for this person from a distance, such as a using a heat camera or possibly some sort of way to track the persons phone, or a chip, if in some radius of the person. Assuming that the people in the search party can communicate with each other they can also share information. Since the person is lost, we can assume that the person is moving relatively randomly. The framework defined in this study could be used to model this problem and our strategies could be used to potentially find the missing person faster. It is possible that the search party has some idea of an area the missing person might be in. It could be a forest, city etc. We can model this area as a grid, where the size is decided by the dimensions of the area, and let it be the arena. Furthermore we can model the search party as a group of cooperating pursuers and the missing person as a single evader moving randomly.

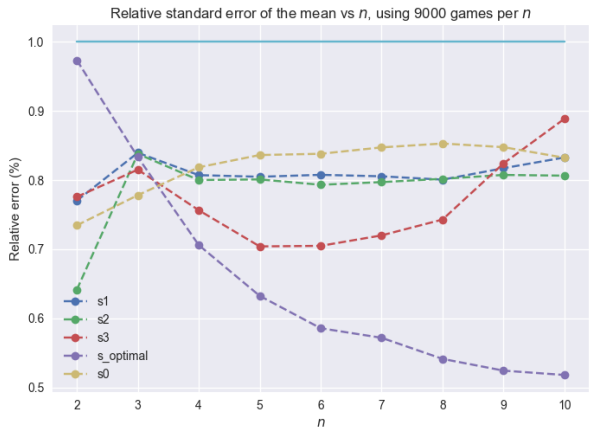


Fig. 13. RSEM versus n graph with $|P| = 2, |E| = 2$ for **Corridor**-type definition of $\mathcal{O}(l_{p_i})$.

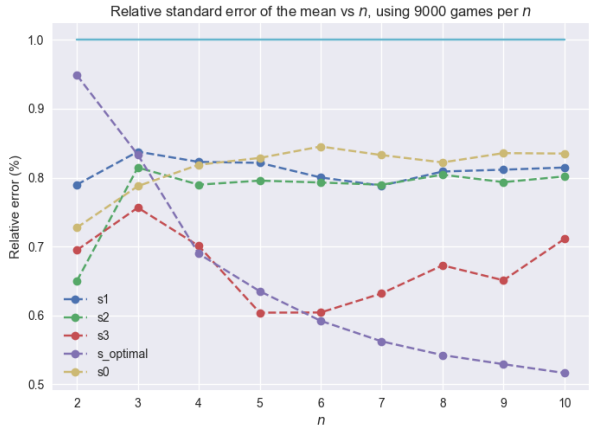


Fig. 14. RSEM versus n graph with $|P| = 2, |E| = 2$ for **Radius**-type definition of $\mathcal{O}(l_{p_i})$.

X. Conclusion

A mathematical framework for Pursuit-Evasion games on grids of imperfect information was defined and implemented in Python. A knowledge representation using matrices and sets of observable locations was constructed and used to find knowledge based strategies. Four strategies were devised, three of which used the knowledge representation and out of these three one was inspired by the Knowledge-Based Subset Construction. The strategies were implemented and simulated in Python. Their performance was then analyzed and compared to each other based on three different criteria: how well the strategies performance scaled with the **1. Size of arena**, **2. Number of Pursuers**, **3. Number of evaders**. The results showed that the strategy inspired by the Knowledge-Based Subset Construction outperformed the other two knowledge based strategies in the first criterion. This can be explained by the fact that this strategy minimizes the uncertainty regarding the locations of the evaders. However, In the second criterion all three knowledge based strategies exhibited very similar performance, and in the third the KBSC-inspired strategy was outperformed by the two others.

XI. Future work

A. Symbolic representation

Our implementation of the KBSC is unfortunately an inefficient implementation for grids of larger sizes. The algorithm for computing $\mathcal{K}$ and $\mathcal{P}$ goes through every element in the matrix-representation and therefore have a time complexity for just the traversal of the knowledge at $\mathcal{O}(n^2)$. To make the algorithm faster for grids of larger sizes, a symbolic representation of the knowledge could be studied. Further work could include studying how the knowledge could be represented by knowledge lists instead of matrices. For instance, we could have four lists telling us where there could be evaders, one for which rows, one for which columns, one for which left diagonals, and one for which right diagonals. This instance would result in the traversal of the knowledge to have a linear time complexity. This area has not been fully explored, but further studies would most certainly yield more effective algorithms.

B. More advanced strategies using the knowledge representation

This study only scrapes the surface on what strategies are able to be constructed using the $\mathcal{K}$ and $\mathcal{P}$ matrices and our framework. One could possibly investigate more advanced strategies involving different strategies for different pursuers or strategies that look multiple turns into the future. Another approach could be to let pursuers communicate what move they are going to make to each other. In this way, coordinated strategies (such as a ”pincer movement”) could be constructed.

C. Different evader behaviours

While analyzing the strategies the evaders were chosen to move randomly. An interesting approach would be to analyze how different evader behaviours affects the performance of the strategies. One potentially interesting behavior to examine is to let the evaders try to flee from the pursuers. This would increase the difficulty for the pursuers enormously and could potentially lead to the pursuers and evaders finding themselves in an infinite loop where the pursuers never manage to catch the evaders. A potential solution to this is to construct strategies where multiple pursuers cooperate to try to corner an evader.

ACKNOWLEDGMENT

The authors would like to thank our supervisor Dilian Gurov for the continuous support and his insightful inputs throughout the project.

REFERENCES

[1] D. Gurov, V. Goranko, and E. Lundberg, "Knowledge-based strategies for multi-agent teams playing against nature," *CoRR*, vol. abs/2012.14851, Feb 2021. [Online]. Available: https://arxiv.org/abs/2012.14851

[2] X. Huang, P. Maupin, and R. van der Meyden, "Model checking knowledge in pursuit evasion games," in *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, Jul 16-22, 2011*, T. Walsh, Ed. IJCAI/AAAI, 2011, pp. 240–245. [Online]. Available: https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-051

[3] L. Doyen and J. Raskin, "Games with imperfect information: theory and algorithms," in *Lectures in Game Theory for Computer Scientists*, K. R. Apt and E. Grädel, Eds. Cambridge University Press, Jun 2011, pp. 185–212.

[4] S. Westerlund and J. Lycken, "Strategy synthesis for multi-agent games of imperfect information," *KTH Bachelor Thesis Project*, Jul 2020, Stockholm.

Multi-Agent Games of Imperfect Information: Algorithms for Strategy Synthesis

David Berisha and Viktor Åkerblom Jonsson

***Abstract*—The aim of this project was to improve upon a tool for strategy synthesis for multi-agent games of imperfect information against nature. Another objective was to compare the tool with the original tool we improved upon and the Strategic Model Checker (SMC). For the strategy synthesis, an existing extension for expanding the games called the Multi-Agent Knowledge-Based Subset Construction was used. The construction creates a new knowledge-based game where strategies can be tested. The strategies were synthesized for the individual agents and then joint profiles of the individual strategies were tested to see if they were winning.**

Four different algorithms for going through the game graphs were tested against the other existing tools. The new and improved tool was faster at synthesizing a strategy than both the old tool and the SMC for almost all games tested. Although for the games where the new tool is out-performed, results indicate it to be due to a combination of chance and how the games are perceived by the tools. No algorithm or tool proved to be the best performing for all games.

***Sammanfattning*—Syftet med detta projekt var att förbättra ett existerande verktyg för att syntetisera strategier för fleragentspel av imperfect information mot naturen. Därefter också jämföra verktyget med original verktyget och med ett verktyg som heter the strategic model checker (SMC). För syntetiseringen av strategier användes ett existerande verktyg för att expandera spel, som kallas Multi-Agent Knowledge-Based Subset Construction. Konstruktionen skapar ett kunskapsbaserat spel där strategierna kan bli testade. Strategierna syntetiserades för de enskilda agenterna och därefter skapades en sammansatt profil av strategier, som då testades för att se om det var en vinnande strategi.**

Fyra olika algoritmer för att gå igenom spelgrafen testades och jämfördes med de andra verktygen. Det nya och förbättrade verktyget var snabbare att syntetisera en strategi än både det gamla verktyget och SMC verktyget för nästan alla spel som testades. Fast, för spelen då nya verktyget inte var snabbast så indikerar resultaten på att detta är p.g.a. en kombination av slump och hur spelen ses på av verktygen. Ingen algoritm eller verktyg visade sig vara det snabbaste för samtliga spel.

***Index Terms*—Strategy Synthesis, Multi-Agent Knowledge-Based Subset Construction, Strategic Model Checker, Multi-Agent Games, Imperfect Information.**

Supervisors: Dilian Gurov

TRITA number: TRITA-EECS-EX-2021:194

I. Introduction

Consider a game represented by a graph and the task of finding a strategy for this game. If the different possible states of the game are represented by states in the graph and the actions that bring the game from one state to another are represented by paths between states, then finding a strategy is quite straightforward. By simply traversing the graph, checking what state the game is in, and what moves brings the player to the desired state, a strategy for the game can be formulated. However, this method is dependent on the complexity of the graph.

If there are several agents playing the game, then there exists a state in the graph for each possible combination of the agents game states, and the possible actions in each state become all the combinations of the separate agents' actions in their respective game state.

In this project, we use a tool by Jacobsson and Nylén [1] to apply the Multi-Agent Knowledge-based Subset Construction as explained in II-D, expanding games of imperfect information into games of perfect information, giving us our graphs representing the game as set states, with the actions connecting them. But instead of searching for strategies directly in the graph for the coalition of agents, we explore the heuristic presented by Gurov et al. [2]. Where strategies for the coalition of agents are found by first searching for memory-less strategies in the graphs for each agent, creating a profile of individual strategies, and then testing this profile of strategies on the game for the coalition of agents.

A tool for synthesizing strategies in this manner has already been created by a prior project group Lycken and Westerlund [3]. Our work is based both on their work and comparison to their results.

In section III-B we present the different algorithms that we explored using for searching for strategies for the individual agents' games.

A. Objectives

The primary objective of this project was to improve upon an existing tool that generates strategy profiles of individual strategies for a coalition of agents and tests these profiles on a multi-agent game of imperfect information that has been expanded so as to be of perfect information.

The secondary objective was to compare the original tool, our tool, and another existing tool called SMC by Pilecki et al. [4]. Where the metrics for comparison and improvement were the time it took the tool to produce a winning strategy and the number of strategies the tool is capable of finding.

B. Delimitation

To find a winning strategy we assume that the game is won when the defined *win-state* is reached and that no further action in the win-state is needed. Another limitation is that we only find winning strategies for a coalition of agents if the

strategy for each individual agent's strategy is a memory-less knowledge-based strategy.
Lastly, we only perform tests on games where each agent only has one singleton observation-state as their start-, win- and lose-state respectively.

II. Background

A. Single-Agent Games

1) With Perfect Information: Games with perfect information can be described by game graphs, which are tuples of the form

$$G = \langle L, l_I, \Sigma, \Delta \rangle$$

L is the finite set of states of the game, l_I is the initial state, Σ is a finite alphabet of actions that can be made at different states and $\Delta \subseteq L \times \Sigma \times L$ are the transitions between the states. Single-player games are played as two-player turn based games. The game starts at the initial state l_I. In subsequent rounds one player will choose an action $\sigma \in \Sigma$ and the other player will decide which state l' the game will transition to, such that $(l, \sigma, l') \in \Delta$. The next round will start in state l'. Calling player two nature means that it has to follow the predefined rules of the game in a non-deterministic way.
In Fig. 1 an example of a single-player game against nature is shown, which can be described by the tuple

$$G = \langle L, l_I, \Sigma, \Delta, \mathcal{O} \rangle$$

Where:

- $L = \{start, one, two, three, four, win, lose\}$
- $l_I = start$
- $\Sigma = (a, b, c)$
- $\Delta = \{(start, b, two), (start, b, one), (start, a, one), (start, c, three), (one, a, lose), (one, b, two), (one, c, two), (two, b, four), (two, c, four), (three, a, four), (three, b, four), (three, b, lose), (three, c, lose), (four, a, win), (four, b, win), (four, c, win)\}$

The $\mathcal{O}$ stands for the set of observations of the game and will be delineated further in the next section about *games of imperfect information*.
In these graphs, the vertices represent the states of the game L and the edges represent the transitions between the states. This can be seen in Fig. 1, where for instance there is a transition between one and $lose$ $(one, a, lose) \in \Delta$. This transition is illustrated in the graph by the edge between those vertices and is labeled by the action a. For the game in Fig. 1, the actions a, b, and c are available at each state. From the graph, it is clear that the same action can lead to different transitions and this is because of the non-determinism where nature decides which transition will occur.
The dashed red line between states one and two displays that these states are indistinguishable to the agent.

2) With Imperfect Information: The example game used in Fig. 1, is a game of *imperfect information*. The game could be defined by the tuple

$$G = \langle L, l_I, \Sigma, \Delta, \mathcal{O} \rangle$$

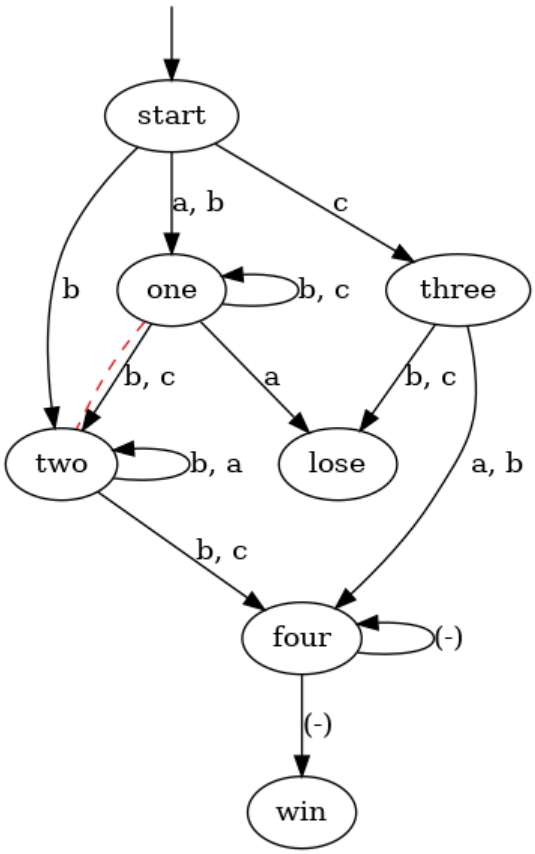


Fig. 1. A single-agent game of imperfect information against nature. The dashed line between the states one and two conveys that they are indistinguishable to the agent. The label (-) on some edges means that all possible actions could lead to that transition.

In this definition, the $\mathcal{O}$ stands for the set of observations $o \in \mathcal{O}$ of the game. This set partitions the set of locations L, this means that for every $l_i \in L$ we have an observation $o(l_i) \in \mathcal{O}$. If two states are indistinguishable to the agent they correspond to the same observation. Thus for the game defined in Fig. 1, we have the observations

- $\mathcal{O} = (o_0(start), o_1(one, two), o_2(three), o_3(four), o_4(lose), o_5(win))$

The observations stands for which state the player observes they are in. The decision of which action he should choose in a given state is based on that observation. Hence if the game is in the state one or two the player would see the game as o_1 and would have to choose its next action not knowing what exact state the game is in. This means that the player will have to chose the same action in both of those states.

B. Multi-Agent Games of Imperfect Information Against Nature

Multi-agent games of imperfect information against nature (MAGIIAN) are defined in a similar way to single-agent games. They are defined by the tuple

$$G = \langle Agt, L, l_I, \Sigma, \Delta, \mathcal{O} \rangle$$

Where $Agt = \{a_1, ..., a_n\}$ is a coalition of n agents. $\Sigma = \Sigma_{a_1} \times ... \times \Sigma_{a_n}$ is the joint action alphabet of the team of

agents, where Σ_{a_i} is the action alphabet of agent i. The agents can not communicate between themselves and will all choose their actions independently before the game transitions into the next state. In Fig. 2 there is an expansion of the game in Fig. 1 to two agents. The transitions in this game are based on the joint actions of the two agents, which can be seen on the labels of the transitions in the graph.

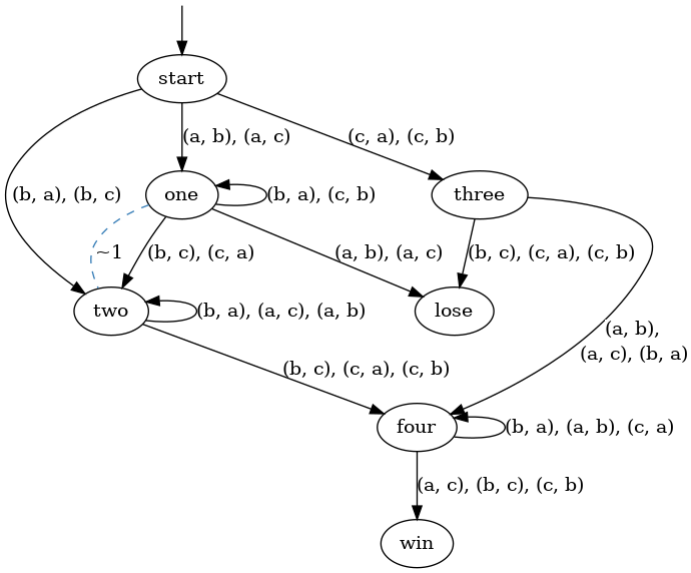


Fig. 2. A multi-agent game of imperfect information against nature. The dashed lined between the states *one* and *two* conveys that they are indistinguishable to agent 1

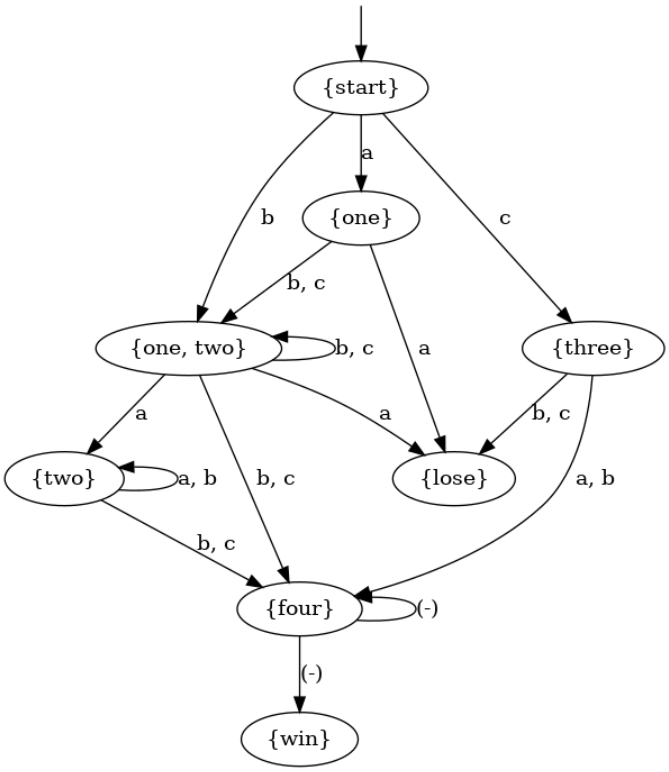


Fig. 3. The game shown in Fig. 1 expanded with the KBSC

C. Knowledge-Based Subset Construction

The *knowledge-based subset construction* (KBSC) [5] is a tool that expands the game to become a game of perfect information. The expanded games are defined by *"knowledge states"* instead of the normal states of the game. These knowledge states represent which states the agent knows the game could be in. In Fig. 3 the KBSC expansion of the game in Fig. 2 is depicted. For the game in Fig. 2 the states one and two were indistinguishable. In the expanded game this indistinguishability is described by the knowledge state $\{one, two\}$, this conveys that the agent knows it is in either of those states. This means that the game has been transformed into a game of perfect information with regard to its knowledge states.

D. Multi-Agent Knowledge-based Subset Construction

The multi-agent knowledge-based subset construction (MKBSC) is a multi-agent extension of the KBSC, introduced in [2]. The generalized construction works as followed.

1) It projects the Game down to all the agents. This creates *n* single-player games of imperfect information.
2) These single-player games are then expanded using the KBSC. This produces *n* single-player games with perfect information.
3) A composition of the games is created, resulting in a single multi-player game with perfect information.
4) Partition the different agents by defining their observations. This finally results in a multi-player game of imperfect information.

If there are winning strategies for the game expanded with the MKBSC then there are winning strategies for the original game, as shown in [2].
In Fig. 4, the MKBSC expanded version of the game in Fig. 2 can be seen. An MKBSC expanded version of a game $\mathbf{G}$ is denoted $\mathbf{G}^{\mathsf{K}}$. In these expanded game graphs the first row in the vertices represents the knowledge state of the first agent, the second row represents the knowledge state of the second agent, and so forth.

E. Strategy synthesis and heuristic

1) Strategies: A strategy is what decides what the next move of an agent should be. To formally define strategies we first define what a play is. A play in a game $\pi = l_0 l_1 ...$ is a infinite sequence, where $l_0 = l_I$ and $\forall i \geq 0 \, \exists \sigma_i \in \Sigma | (l_i, \sigma_i, l_{i+1}) \in \Delta$, as defined by Doyen et al. [6]. From a play, one can define a history as a finite prefix of a play $\pi(i) = l_0, ..., l_i$. With this, a deterministic strategy can be defined as functions that map histories to actions $\alpha : L^+ \rightarrow \Sigma$. A strategy is said to be *memoryless* if it only depends on the last location of the history, according to [6].
Some strategies are called *surely winning* strategies, such strategies guarantee that the player will win the game if the strategy is followed. There are also *almost surely winning* strategies, and these strategies will sometimes lead to a win but they can not guarantee it.

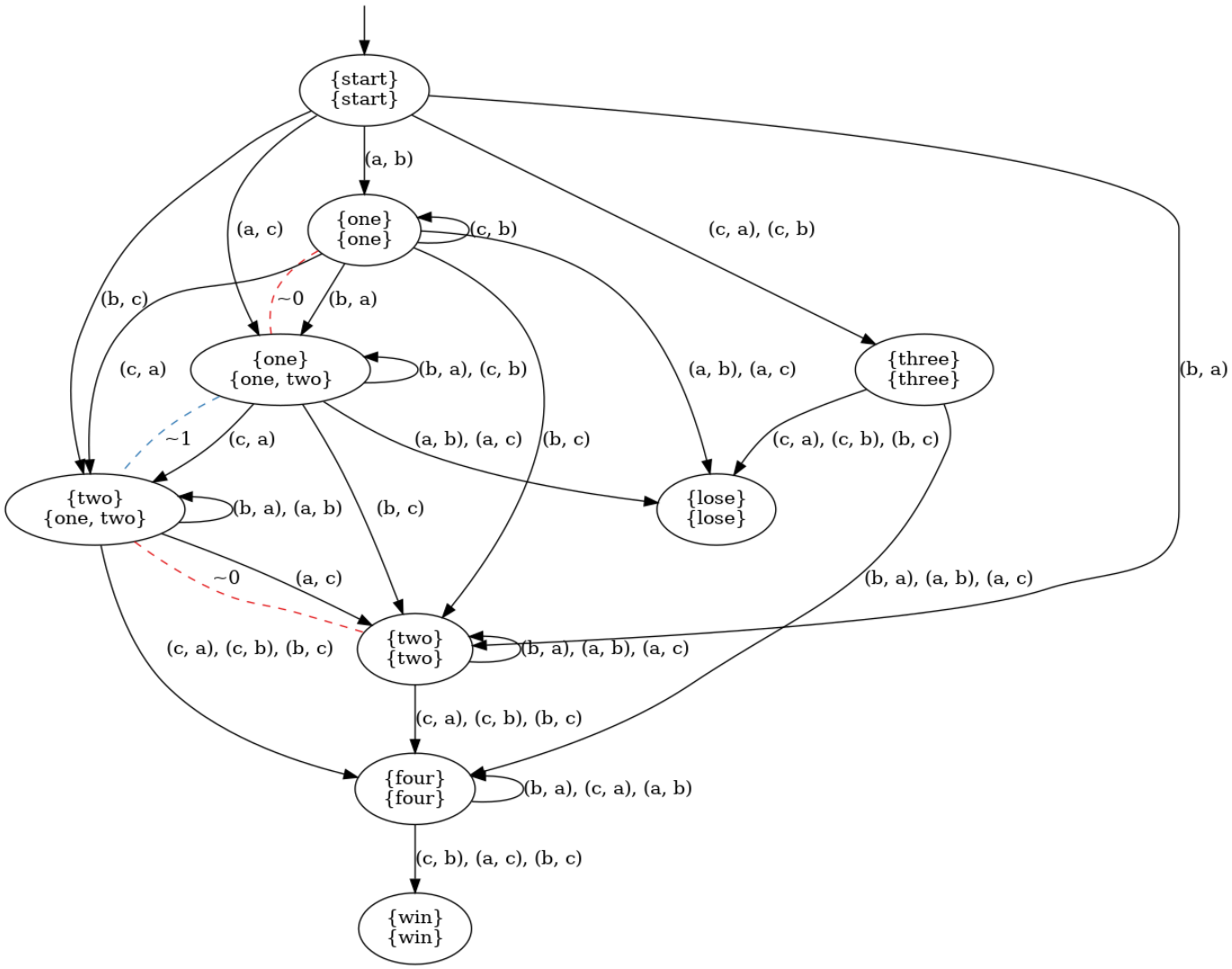


Fig. 4. The game shown in Fig. 2 expanded with the MKBSC

2) Objectives: Games can be modelled with many different types of objectives. The games tested in this project all had *reachability objectives* and thus we will only cover this type of objective. We use the definition of reachability objectives described in [6]. With a set of target locations $\mathcal{T} \in L$, the reachability objective written $\mathbf{Reach}(\mathcal{T}) = \{l_0 l_1 ... | \exists k \geq 0 : l_k \in \mathcal{T}\}$ is achieved if an observation state in $\mathcal{T}$ is visited at least once in a play. For clarity the reachability objectives for the games in this report are the observations called win, however, the tool can handle other reachability objectives.

3) Heuristic: The heuristic presented by Gurov et al. [2] to synthesize profiles of observation based memoryless strategies in $\mathbf{G}^{\mathsf{K}}$:

- For all the games where $\mathbf{G}$ has been projected down to all the agents i and then expanded with the KBSC $(\mathbf{G}|_i)^{\mathsf{K}}$, find memoryless strategies α_i^{K} that have a winning outcome for the reachability objectives.
- Check whether the strategy profile $\{\alpha_i^{\mathsf{K}}\}_{i \in Agt}$ is winning for the reachability objective for the MKBSC expanded game $\mathbf{G}^{\mathsf{K}}$.

If there is a profile of winning observation-based perfect recall strategies in $\mathbf{G}^{\mathsf{K}}$ for its translated reachability objective R^{K}, then there is also one in $\mathbf{G}$ for the reachability objective R, according to [2]. Where R^{k} is the translated reachability objective for the MKBSC expanded game.

F. SMC

The *strategic model checker* (SMC) is a tool designed by Pilecki et al. [4] for model checking and synthesizing strategies. It works for a subset of alternating-time temporal logic (ATL) with imperfect information and imperfect recall.

III. Method

A. Strategy synthesis

The method for synthesizing strategies is primarily divided into three steps. The first step consists of using an algorithm to generate one or a set of winning strategies for each agent. The

second step is to test each untested combination of the agents' strategies on the game for the coalition of agents. Finally, the third step is to take the already generated strategies for one agent, and generate new untested strategies using the already generated ones.
Steps three and two are thereafter repeated until a winning strategy for the coalition of agents has been synthesized. See Appendix A for pseudo-code of the method for strategy synthesis.

B. Algorithms

The following algorithms were tested for synthesizing strategies. See Appendix B for pseudo-code of the algorithms.

1) Full forwards search: To generate the initial set of winning strategies for each agent, a breadth-first search algorithm is applied to the agent's graph from the starting state. As the algorithm traverses the game, it saves each state it visits along with one of the possible actions in that state that does not go to itself or a lose-state. The set of these states and their respective actions is considered a strategy for the agent. If more than one action is relevant in any one state that is visited, a copy of that state, along with each action is saved separately. These states and their coupled actions are hereafter referred to as *conflict-states*. When the search has exhausted the graph, the saved states make up a strategy for the agent.
For example, by applying this algorithm to the graph in Fig. 5, one could generate a strategy similar to the one illustrated

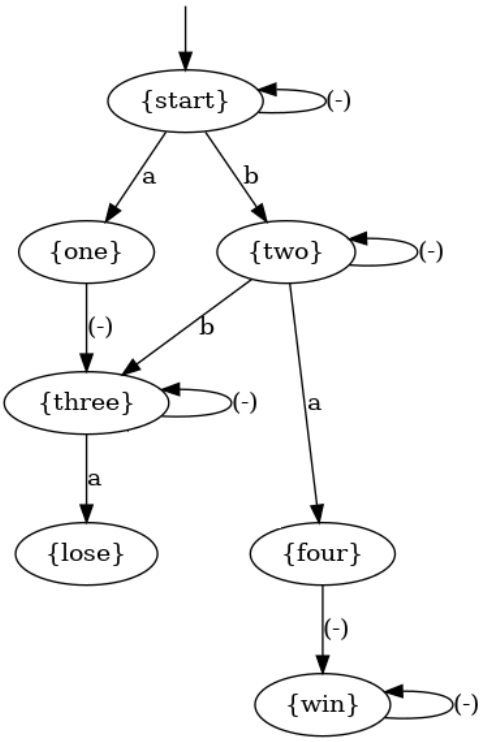


Fig. 5. Example graph to help illustrate algorithm in III-B1 and III-B2

in Fig. 6 with conflict-states according to Fig. 7.
Creating new strategies in the third step is thereafter trivial,

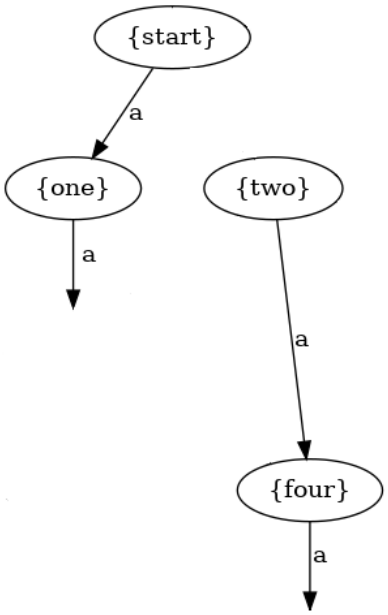


Fig. 6. Example strategy that could be generated using the algorithm in III-B1

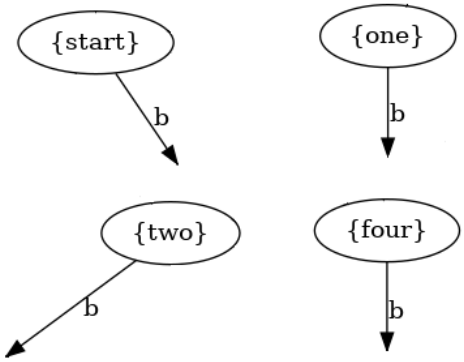


Fig. 7. Example of the conflict-states coupled with the strategy presented in Fig. 6

as the breadth-first search visits all non-lose-states that can be visited between the start and win-state and the conflict-states contain all alternate actions that can be taken in these states. Therefore, to generate new strategies for the agent one simply has to take the existing set of strategies for the agent, duplicate it, modify the duplicate set using one of the states and one of the actions coupled to that state, and then add the modified duplicate set to the set of existing strategies for the agent. The action is then removed from the state in the set of conflict-states after use, and any state in the set of conflict-states without coupled actions is removed from the set.
One major drawback with this algorithm is that, while it will generate all winning strategies for the agent, it won't only generate winning strategies, as is shown by Fig. 6, which is not a winning strategy.
Another drawback is it can generate redundant strategies. An example of how this could happen can be shown if you compare the strategy in Fig. 6 and the conflict-nodes in Fig.

7. If the move in {*start*} is set to *b*, then the move in {*one*} is redundant. Despite this, the algorithm will generate a strategy for each move in {*one*} with the move in {*start*} set to *b*.

2) Full backwards search: This algorithm is the same as the one presented in III-B1 but instead of traversing the game from the start-state to the win-state, the algorithm traverses the game inversely from the win-state to the start-state.
Applying this algorithm to the same graph as the forwards algorithm Fig. 5 would result in generating a strategy similar to Fig. 8. The conflict-states would then be according to Fig. 9.

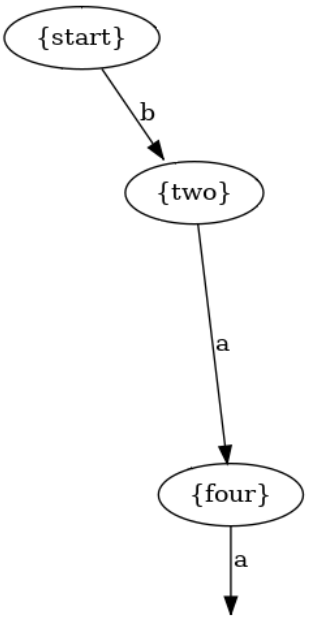


Fig. 8. Example strategy that could be generated using the algorithm in III-B2

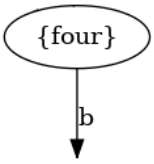


Fig. 9. Example of the conflict-states coupled with the strategy presented in Fig. 8

The benefit of this algorithm is that as it explores the graph from the win-state to the start-state, only states that may be visited during winning strategies are included in the strategy. Therefore only winning strategies will be generated, and the amount of redundant strategies may be reduced as the amount of states to modify is reduced.

3) Partial forwards search: Just like in III-B1, a breadth-first search is implemented to generate strategies. But instead of initially generating only one strategy from a start-state to a win-state, a set of strategies is generated. Whenever the graph splits, so that one can traverse from one state to several different, all states are not simply added to the strategy being generated. Instead, a duplicate of the strategy is created so that there is one for each possible state, thereafter, one of the possible states is added like normal to each respective strategy. This means that when the breadth-first search is exhausted, the set of winning strategies for the agent will be equivalent to the set of different paths along which one can traverse the graph from the start-state to a win-state when playing the game. The strategies that only cover one path through the game are hereafter referred to as *partial strategies*
Also, each state in the set of conflict-states is now also coupled to a certain partial strategy for the agent. Meaning that instead of modifying all strategies for the agent when generating new ones, only the ones coupled to the state in the set of conflict-states are modified.
To clarify this, when the algorithm is applied to the graph in Fig. 10. Two strategies will initially be generated, the

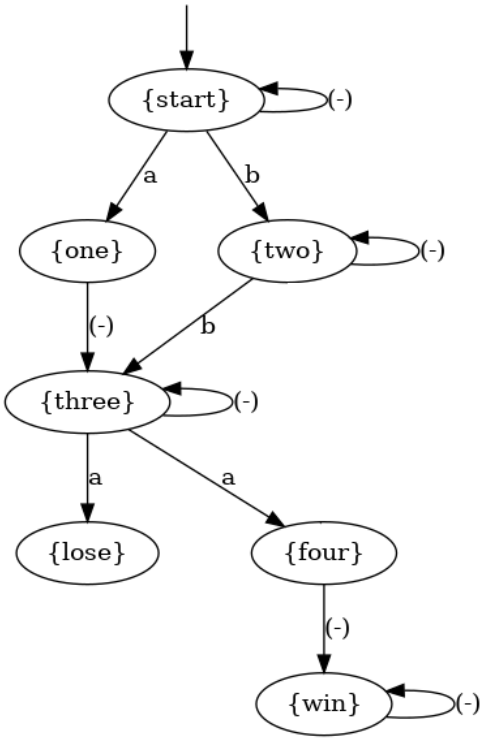


Fig. 10. Example graph to help illustrate algorithms in III-B3 and III-B4

one presented in Fig. 11 coupled with the conflict-states in Fig. 12, and the one presented in Fig. 13 coupled with the conflict-states in 14.
The benefit of this is that, as the partial strategies are linear paths between the start-state and win-state, using conflict-states to generate new strategies might be less prone to generate redundant copies of strategies for the agent.
The drawback however is that as the strategy only processes a limited amount of states, it produces a loss whenever it's presented with a state outside its scope, however winnable the game is from that state.

4) Partial backwards search: This algorithm is the same as the one presented in III-B3 with the exception that just like in

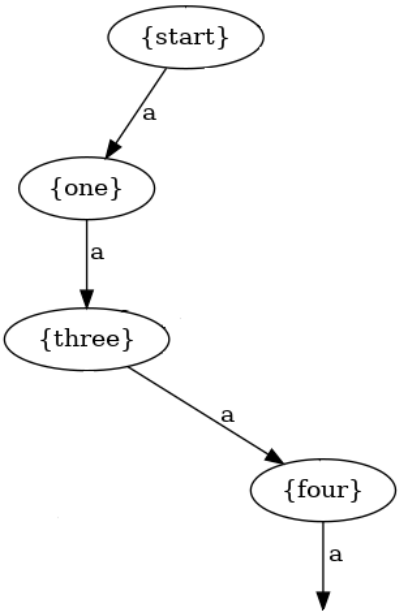


Fig. 11. Example strategy that could be generated using the algorithm in III-B3

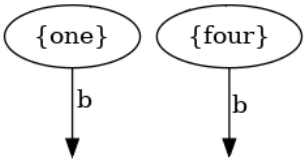


Fig. 12. Example of the conflict-states coupled with the strategy presented in Fig. 11

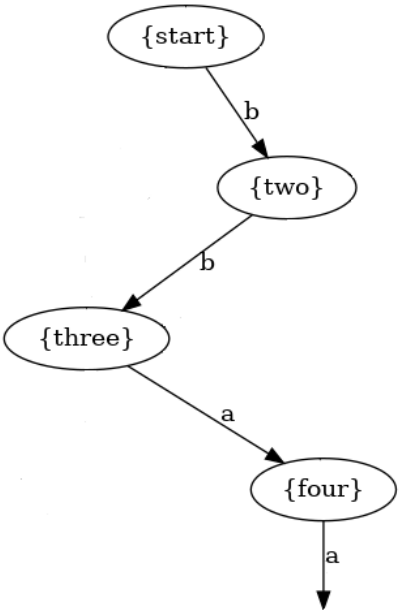


Fig. 13. Example strategy that could be generated using the algorithm in III-B3

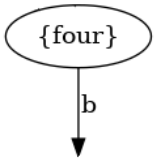


Fig. 14. Example of the conflict-states coupled with the strategy presented in Fig. 13

III-B2 the algorithm traverses the game inversely, which in this case generates strategies consisting of the same set of states. While using a backwards search does not provide the benefit of ignoring redundant states in the final strategy as this is already achieved by only saving the direct paths through the graph, the algorithm still benefits from never visiting the redundant states while searching through the graph in the initial phase.

C. Testing

The algorithm for testing a strategy of a coalition of agents is quite straightforward. The first step is taking a state, which initially is the relevant start-state for the coalition of agents game-graph, and checking what profile of actions the coalition of agents wishes to take in this state. This profile of actions is compared to the game to see what states one may traverse to using it. If this set of states contains a state already visited during the test or a lose-state, the strategy is marked as having lost. The first step of the algorithm is then performed once again on the remaining states. If there are no such states, then the strategy is also marked as having lost. Note that as the algorithm compares the state to all visited states, and not just the state on that path, the algorithm sometimes miss-classifies surely winning strategies as winning strategies. However, if a win-state exists in the produced set of states, the strategy is marked as having won. When the set of possible states to visit is exhausted, any strategies that are marked as having won are saved as winning or surely winning depending on whether or not they are also marked as having lost. See Appendix C for pseudo-code of the testing method.

D. Comparing with other tools

1) Speed: To compare the tools' effectiveness at finding a strategy in relation to time spent searching, the Python3 library *Timeit* was used and the time taken for our tool and the original tool to return at least one winning strategy was measured over 1000 loops. To give an as fair comparison as possible to the SMC tool, the entire process was timed, from the tool being given the expanded game to it finishing its task.

This is however an unfair comparison as our tool parses and stores the data from the file containing the expanded game differently from how the tool by Lyckén and Westlund [3] does it, as well as being capable of terminating after finding one strategy. For this reason, a modified version of their algorithm for generating strategies for agents was implemented in our tool. This algorithm is explained in III-B3.

When measuring the time taken for the SMC tool to return a

strategy, two modifications to the method were implemented. One was that as the game given to the tool was first converted from the expanded form generated by Jacobsson and Nyléns tool [1] to a format the SMC tool could handle using a tool we developed which can be found at Github[1].
The other modification was that instead of using Timeit to measure the time taken by the tool, the time reported by the tool itself was referred to when measuring the time taken to synthesize a strategy. The motivation behind this work-around being the SMC-tool being a .jar file and not python3 code.

2) Amount of strategies found: The focus was primarily on the algorithms for our tool when comparing the number of strategies found. The reason for this being the SMC tool being designed to only find one strategy, and the algorithm for the old tool having been implemented in our tool.

E. Games Tested

As our focus was on the improvement of the tool Lycken and Westerlund [3], most of the games tested were taken from that report, and the last game presented here is the only exception.

1) Chemical Game: As presented in Fig. 15, the *Chemical Game* is a game played by two robots. At the start they can only choose the action grab. When this is done, their collective grip is either *good* or *bad*, but only the first robot can observe which is the case. To win the game, the robots must both choose to *lift* while their grip is *good*. Failure and loss of the game can be achieved if the robots both choose to *lift* while their grip is *bad* or by the robots not choosing the same action. While this game was taken from the report by Lycken and Westerlund [3], they had sourced it from the report by Jacobsson and Helmer [1].

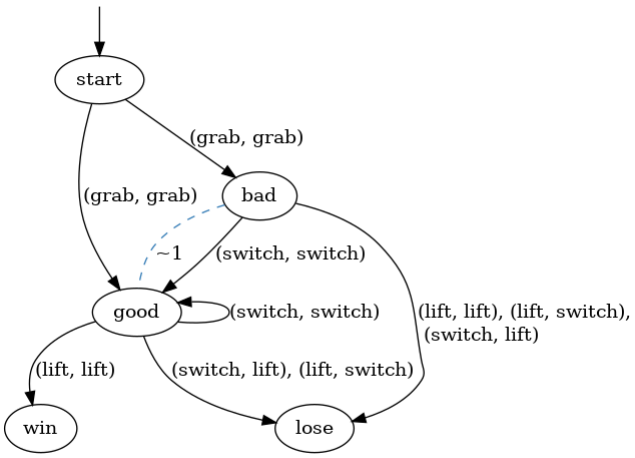


Fig. 15. Graph of the *Chemical Game* in it's original form

[1]https://github.com/VAkerJ/ISPLconverter

2) Supervisor Game: The *Supervisor Game*, named so as it was given to the previous group by their supervisor [3], does not test any particular function of our tool and is presented in Fig. 16. It does however have an interesting feature that even when expanded, the first agent may never reach an observation-state where they know that the game is in state *one*.

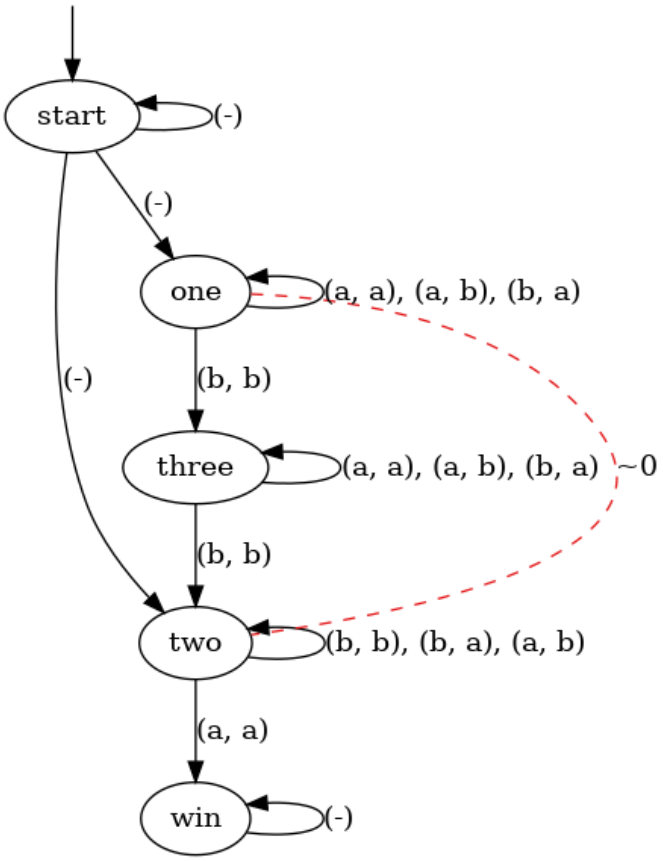


Fig. 16. Graph of the *Supervisor Game* in it's original form

3) Almost-/Surely Winning Game: The *Almost-/Surely Winning Game* which is presented in Fig. 17, is a game constructed in order to test the tools' ability to synthesize both strategies that are guaranteed to win the game and strategies that only win the game sometimes. As this is the case, it is a game of perfect information and does not change when expanded.

4) Looping Game: Just like in III-E3 the *Looping Game* seen in Fig. 18 is a game of perfect information. This is due to the game only being used to test the tools' ability to handle games that can loop back to previously visited game-states.

5) Cops and Robbers Game: The graph for this game is too complicated to fit in this report, therefore Fig. 19 displays the expanded game from the perspective of one of the agents playing, which is identical for all agents.
The basis of the game is that a robber or *thief* hides in one of four rooms, the thief in this scenario can be considered nature and the game-state is what room the thief is hiding in.

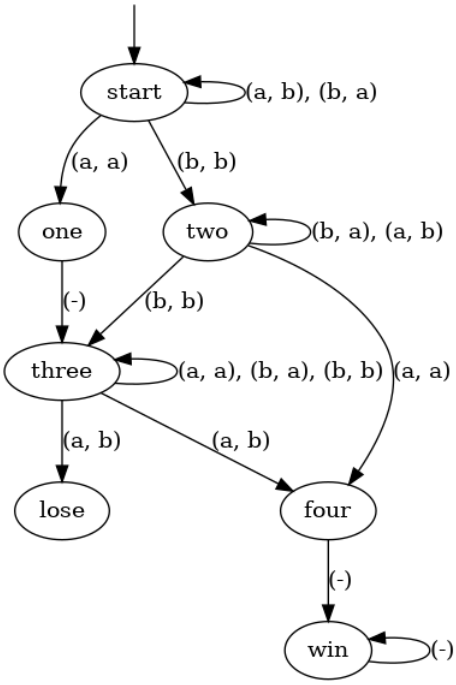


Fig. 17. Graph of the *Almost-/Surely Winning Game* in it's original form

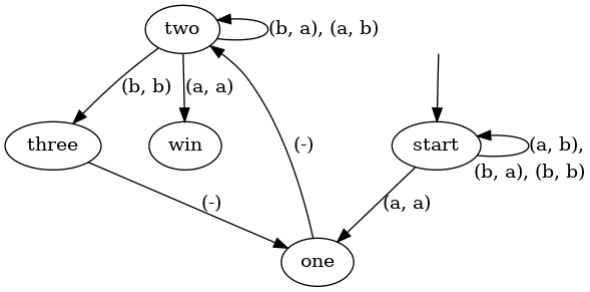


Fig. 18. Graph of the *Looping Game* in it's original form

The agents playing the game are cops and their action each turn is to choose a room to check. If a cop checks the room the thief is in, the game is won If they don't check the room the thief is in, the thief moves to one of the adjacent rooms. In this case, room one is next to two, two is next to one and three etc.
In Fig. 19, the observation-states are the rooms the thief could be in, and the actions are the rooms the cop chooses to check. Note that since only one cop needs to find the thief, and all rooms can be checked from each observation-state, all actions in every observation-state might lead to a win.

6) Triple Agent Game: The *triple agent game* can be seen in Fig. 20. The game was created to test how the different tools could handle games with more than two agents that is of imperfect information. The game consists of three agents that all have three actions available to them at each given state.

TABLE I
THE TIME (T) IN MILLISECONDS TAKEN FOR OUR TOOL UTILIZING THE FULL FORWARD SEARCH (FFS), FULL BACKWARDS SEARCH (FBS), PARTIAL FORWARDS SEARCH (PFS), AND PARTIAL BACKWARDS SEARCH (PBS) ALGORITHM TO FIND AT LEAST ONE WINNING STRATEGY FOR EACH GAME, COMPARED TO THE PERFORMANCE OF THE ORIGINAL TOOL AND THE SMC TOOL. THE AMOUNT OF WINNING STRATEGIES RETURNED (NR. FOUND) WHEN AT LEAST ONE HAD BEEN FOUND IS ALSO INCORPORATED FOR CONTEXT.

Algorithm/ tool	Metric	Chem.	Sup.	Alm./Sure.	Loop.	C.&R.	Triple.
FFS	T[ms]	2.26	1.78	1.75	1.88	20.0	237
	nr. found	1	1	1	1	1	72
FBS	T[ms]	1.51	1.80	1.66	1.29	19.8	997
	nr. found	1	1	1	1	1	256
PFS	T[ms]	2.52	2.73	4.18	1.40	737	50.8
	nr. found	3	3	3	1	121	3
PBS	T[ms]	2.43	2.70	3.88	1.38	730	54.2
	nr. found	3	3	3	1	121	2
Original	T[ms]	2.98	2.85	3.07	2.49	15.4	N/A
	nr. found	3	3	3	1	100	N/A
SMC	T[ms]	25.1	13.4	27.9	28.7	184	58.3
	nr. found	1	1	1	1	1	1

TABLE II
THE NUMBER OF WINNING (W) AND SURELY WINNING (SW) STRATEGIES FOUND WHEN SEARCHING FOR ALL POSSIBLE STRATEGIES IN THE GAMES. A COMPARISON OF OUR TOOL USING THE FULL FORWARD SEARCH (FFS), FULL BACKWARDS SEARCH (FBS), PARTIAL FORWARDS SEARCH (PFS), AND PARTIAL BACKWARDS SEARCH (PBS) ALGORITHMS WITH THE ORIGINAL TOOL AND THE SMC TOOL.

Algorithms	W/SW	Chem.	Sup.	Alm./Sure.	Loop.	C.&R.	Triple.
FFS	W	5	2	5	0	N/A	0
	SW	0	0	1	1	N/A	1413120
FBS	W	2	2	5	0	N/A	0
	SW	0	0	1	1	N/A	34816
PFS	W	3	3	2	0	5576	0
	SW	0	0	1	1	0	99
PBS	W	3	3	2	0	5576	0
	SW	0	0	1	1	0	99
Original	W	3	3	2	0	68	N/A
	SW	0	0	1	1	32	N/A
SMC	W	1	1	0	0	1	0
	SW	0	0	1	1	0	1

IV. RESULTS

A. Full forwards search

From Table 1, the results for FFS show that the algorithm was the second-fastest algorithm for the first three games, being very close to the fastest algorithm for the second and third games. For the triple agent game, it was the second slowest algorithm. This could be related to the fact that when searching for a minimum of one strategy, it returned the second-largest amount of all algorithms. From Table 2, the FFS algorithm found fewer strategies than FBS for the chemical game and the triple agent game when searching for one strategy.
When searching for all possible strategies, FFS found as many or more strategies for five of the games. Particularly for the triple agent game where it found far more than any other algorithm.
No results for the total amount of strategies found for the cops and robbers game is given, as the computer these tests were performed on didn't have enough RAM to complete this task.

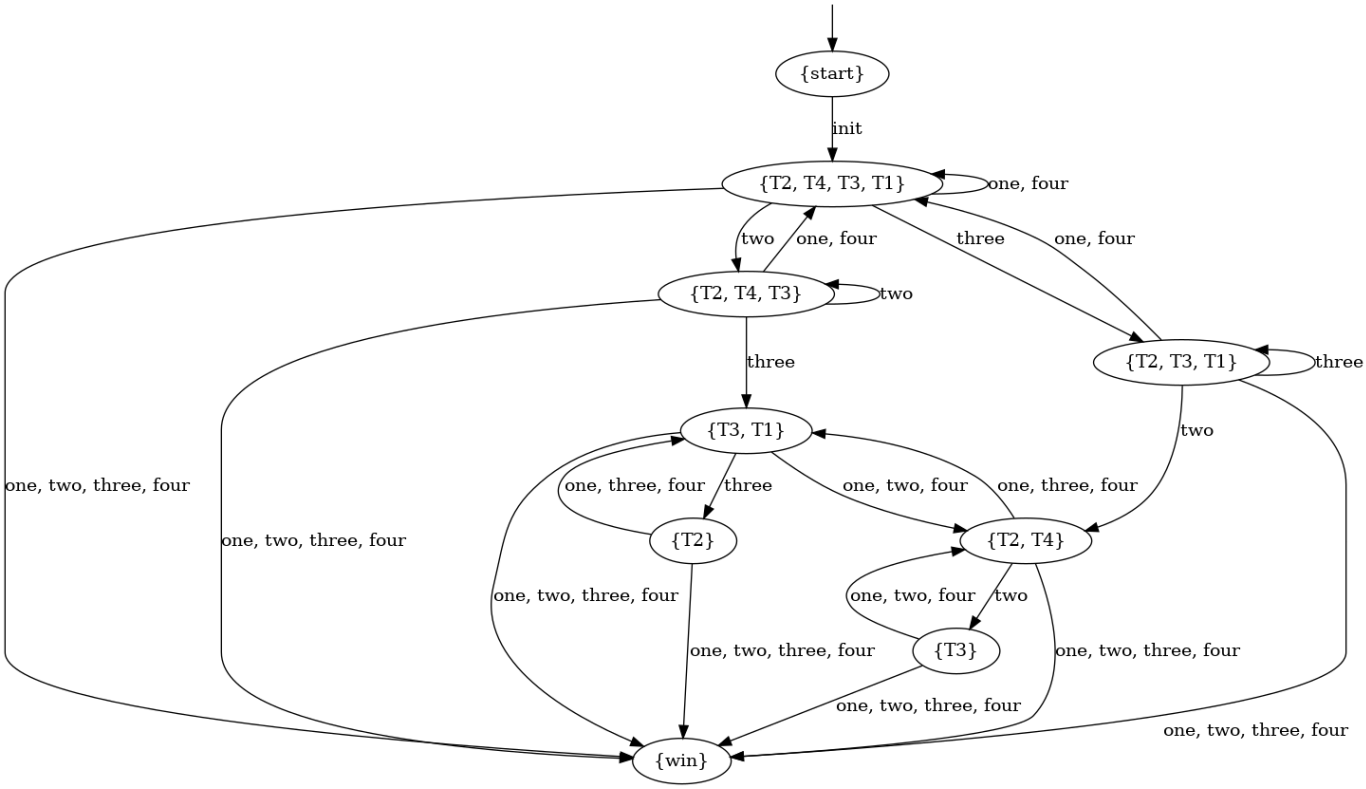


Fig. 19. Graph expanded *Cops and Robbers Game*, as perceived by one of the agents playing.

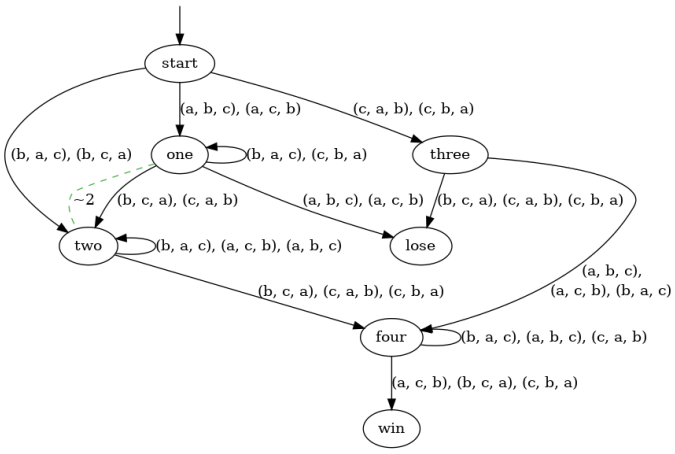


Fig. 20. Graph of the *Triple Agent Game* in it's original form

B. Full backwards search

From the results Table 1, one can see that FBS found a strategy faster than all other algorithms for the first 5 games. For the sixth game however, the *triple agent game*, it was the slowest algorithm at finding a strategy. This might, in a similar way as to the FFS, be related to the fact that the algorithm finds 256 strategies for the triple agent game, which was substantially more than the other algorithms.

No results for the total amount of strategies found for the cops and robbers game is given, as the computer these tests were performed on didn't have enough RAM to complete this task.

C. Partial forwards search and Partial backwards search

Partial forwards search and partial backwards search had very similar results. They found the same number of strategies for all games which was expected. For most games, they found fewer strategies than the other algorithms while searching for all strategies. For all games except the triple agent game, the PBS was a little faster than the PFS as it searches through the game graphs faster. Both of these were noticeably slower than PBS and PFS for all games except the triple agent game and the Loop game.
Although, these algorithms were only tested on games where each agent only has one possible starting observation-state and only one winning observation-state. Changing this might tip the balance between how these algorithms performed.

D. Tool by Lyckén and Westerlund

The tool created by Lyckén and Westerlund is slower in general than the other tools except for the cops and robbers game. The tool could not deal with the triple agent game at all. This is probably because of a quirk in how the tool processes its data which simplifies the agents' graphs but sometimes oversimplifies them. This could also be the reason for the tool being the fastest at finding a strategy for the cops and robbers game. What suggests this is that the tool finds a lot fewer strategies for that game than the PFS which uses the same algorithm to search through the game graph, indicating that something is not working as expected.

E. SMC

The SMC tool was much slower than all the other tools and algorithms for the first four games. For the sixth game, it was faster than the FFS and FBS and almost as fast as the PFS and PBS. The SMC tool only produces one strategy.

V. Analysis

A. Algorithms and tools

The different versions created were overall faster than both the SMC and the tool created by Lyckén and Westerlund. The cops and robbers game and the triple agent game accounted for the biggest inconsistencies and differences between the tools. These games were a bit more complex than the other games and are probably the reason for these games diverging in their results. The fastest algorithm overall is the full backwards search, even though it was the slowest algorithm for the triple agent game. This game displays the weakness of the FBS the clearest. The fact that it returns such a large amount of strategies when searching for only one, indicates that the tool spent a lot of time generating similar and/or redundant strategies. Before finally stumbling upon one strategy for one agent that can be combined with the previously generated strategies to create a winning one.
It is difficult to compare with the original tool completely, as it has questionable results for the last two games. Hence a comparison with the PFS is fairer as it uses the same search algorithm but has more expected results for the last two games. Here the FBS is faster for the cops and robbers game and slower for the triple agent game. The SMC tool overall seemed to be quite consistent as it always only found one strategy but it was overall slower than the other tools. The triple agent game added a lot of complexity which can clearly be seen in the results. This indicates that with more agents the games become much more complex.

B. The tool

The tool has been improved upon in the manner that it is more modular, parses data more efficiently, and stores data in a more optimized manner, which is pointed out by Lyckén and Westerlund in their report [3] to be valuable contributions to their tool. Although this statement could be contested, as the new tool sometimes perceives games as more complex than the old tool due to actions taken to avoid oversimplification of the agents' graphs.
The tool would however benefit from the improvements mentioned in V-C2 and V-C3, as well as being converted into a lower-level language than Python3.
Access to the tool can be achieved Github[2].

C. Future work

1) Comparing to searching in the expanded game: The basis for this tool is the heuristic that finding strategies by first generating them for the individual agents simpler graphs is faster than directly generating for the complex graph for the coalition of agents. Therefore it would be interesting to see how long it would take to find strategies for complex games in using our tool compared to directly applying the algorithms to the graph for the coalition of agents.

2) Generate algorithms while testing: The current algorithm for testing is quite simple. It takes a strategy, tests it on the game, and returns the results. This results in a lot of unnecessary computation, as when new strategies are generated, they are tested as if no other tests have been performed and the algorithm starts traversing the game from the start-state. Meaning that if the algorithm has to traverse the game to where the strategy has been modified before any new information can be gained, and if the modification is in a state not visited during previous tests, then the entire test is implicitly redundant.
A more effective test function would either start traversing the game from where the latest change to the algorithm has been implemented to avoid computations where the answer is already known. Or it could generate modifications to the strategy being tested when the game is not traversed as would be expected.
Implementing this algorithm, while seemingly complicated and possibly impossible, would likely drastically improve the speed of the tool and would be an interesting development.

3) Simplify the games when parsing data: Sometimes the move an agent makes does not matter for the coalition but which matters for the agents game, and sometimes it matters for the agents game but not the coalition. If several actions for an agent in the expanded game and in the agents game are interchangeable, generating several strategies for the actions results in unnecessary computations. Therefore, if these interchangeable actions could be considered one action, the graphs would be simpler, while no information would be lost.

VI. Conclusion

We improved upon a tool for strategy synthesis of multi-agent games of imperfect information, expanded by the Multi-Agent Knowledge-Based Subset Construction. The tool was then compared to the original tool we improved upon created by Lyckén and Westerlund, and with the Strategic Model Checker. We can draw the following conclusions:

- The tool created is overall faster at synthesizing strategies than the original tool and the SMC, with some exceptions. Different results might be achieved if the focus is put on more complex games.
- A lot more complexity is added to the games with more agents.

Appendix A
Pseudo code for strategy synthesis

Appendix B
Pseudo code for agent strategy generation

Appendix C
Pseudo code for strategy testing

Acknowledgment

We would like to thank our supervisor Dilian Gurov for teaching us about game theory and for his helpful advice on

[2]https://github.com/VAkerJ/strategysynthesiser_2

our project. We would also thank the previous group of Simon Westerlund & Jakob Lyckén for their tool, which our project is based on, and August Jacobsson & Helmer Nylén for the tool without which this project would've been a lot more difficult. We would like to formally thank the creators of the SMC for giving us access to the tool and offering up their help in using their tool.
Finally, we would like to thank K. Jonsson & F.D.C. Willard for their useful contributions to the discussion concerning software engineering.

REFERENCES

[1] A. Jacobsson and H. Nylén, "Investigation of a knowledge-based subset construction for multi-player games of imperfect information," Bachelor's thesis, KTH, Stockholm, May 2018.
[2] D. Gurov, V. Goranko, and E. Lundberg, "Knowledge-based strategies for multi-agent teams playing against nature," *CoRR*, vol. abs/2012.14851, 2020. [Online]. Available: https://arxiv.org/abs/2012.14851
[3] J. Lycken and S. Westerlund, "Strategy synthesis for multi-agent games of imperfect information," Bachelor's thesis, KTH, Stockholm, Jul. 2020.
[4] J. Pilecki, M. Bednarczyk, and W. Jamroga, "Smc: Synthesis of uniform strategies and verification of strategic ability for multi-agent systems," *Journal of Logic and Computation*, vol. 27, pp. 1871–1895, Sep. 2017.
[5] J. H. Reif, "The complexity of two-player games of incomplete information," *Journal of Computer and System Sciences*, vol. 29, no. 2, pp. 274–301, Dec. 1984.
[6] L. Doyen, J.-F. Raskin, and E. Cachan, "Games with imperfect information: Theory and algorithms," *Lectures in Game Theory for Computer Scientists*, Jan. 2011.

En spelteoretisk AI för Stratego

Giorgio Sacchi och David Bardvall

***Abstract*—Many problems involving decision making with imperfect information can be modeled as extensive games. One family of state-of-the-art algorithms for computing optimal play in such games is Counterfactual Regret Minimization (CFR). The purpose of this paper is to explore the viability of CFR algorithms on the board game Stratego. We compare different algorithms within the family and evaluate the heuristic method "imperfect recall" for game abstraction. Our experiments show that the Monte-Carlo variant External CFR and use of game tree pruning greatly reduce training time. Further, we show that imperfect recall can reduce the memory requirements with only a minor drop in player performance. These results show that CFR is suitable for strategic decision making. However, solutions to the long computation time in high complexity games need to be explored.**

***Sammanfattning*—Många beslutsproblem med dold information kan modelleras som spel på omfattande form. En familj av ledande algoritmer för att beräkna optimal strategi i sådana spel är Counterfactual Regret Minimization (CFR). Syftet med denna rapport är att undersöka effektiviteten för CFR-algoritmer i brädspelet Stratego. Vi jämför olika algoritmer inom familjen och utvärderar den heuristiska metoden "imperfekt minne" för spelabstraktion. Våra experiment visar att Monte-Carlo-varianten External CFR och användning av trimning av spelträd kraftigt minskar träningstiden. Vidare visar vi att imperfekt minne kan minska algoritmens lagringskrav med bara en mindre förlust i spelstyrka. Dessa resultat visar att CFR är lämplig för strategiskt beslutsfattande. Lösningar på den långa beräkningstiden i spel med hög komplexitet måste dock undersökas.**

***Index Terms*—Counterfactual Regret Minimization, AI, Imperfect recall, Wargames, Imperfect infomation games, Stratego.**

Supervisors: Mika Cohen and Farzad Kamrani

TRITA number: TRITA-EECS-EX-2021:195

I. Introduktion

De senaste årens utveckling av självlärande AI har i grunden förändrat hur abstrakta strategispel bedrivs och kan bedrivas. Ett abstrakt strategispel kan med en måttlig arbetsinsats ges en AI som på egen hand lär upp sig själv i spelet genom att spela mot sig själv om och om igen, som exempelvis AlphaZero demonstretat [1]. Det här innebär att det nu är praktiskt möjligt att utveckla AI även för nischade abstrakta strategispel, som exempelvis de specialiserade datorkrigsspel som används inom försvarsmakter runt om i världen för att utveckla och träna militär taktik och strategi. Datorkrigsspel kom till användning 1990 då Pentagon hastigt behövde sammanställa en plan för att besvara Kuwait-invasionen. Man utvecklade då strategier genom att spela Gulf Strike [2].

Datorspelare i krigsspel har dock länge varit svaga; de har krävt mycket domänkunskap för att implementera och gick ändå ofta att överlista. Nya algoritmer visar däremot stor förbättring på båda punkter. Google DeepMinds berömda algoritm AlphaZero kunde appliceras på en rad strategiska brädspel som schack, go och shogi där den kunde nå övermänsklig prestanda [1].

I och med detta har Totalförsvarets forskningsinstitut (FOI) fått ökat intresse för användningen av algoritmer för strategiframtagning. Stratego bygger på dold information som tillåter spelare att bluffa och på olika sätt lura sin motståndare medan de försöker extrahera information från ens beteende, vilket på många sätt efterliknar militära situationer. Dessutom har Stratego relativt få specifika regler och kan anpassas för olika ändamål med simpla regeländringar. Utöver detta är Stratego från ett AI-perspektiv relativt outforskat och i dagsläget saknas starka AI-spelare. Allt detta gör FStratego till ett intressant och potentiellt lärorikt forskningsområde inom algoritmisk strategiframtagning. Detta arbete kommer utforska strategiframtagning för Stratego med AI-algoritmfamiljen Counterfactual Regret Minimization (CFR). I dagsläget används CFR-varianter i de främsta algoritmerna för spel av imperfekt information så som ReBeL [3] och DeepStack [4].

A. Problemformulering

Arbetet har som mål att undersöka hur effektivt en AI-spelare kan tränas att spela nedskalad Stratego. Grundalgoritmen CFR är dock inte särskilt effektiv på komplexa spel då den kräver lång beräkningstid och stort arbetsminne av datorn den körs på. Arbetet undersöker därför lösningar som påskyndar träningen i form av algoritmvarianter inom kategorin "Monte-Carlo CFR" och trimning, samt lösningar som minskar minnesanvändningen i form av det heuristiska tillägget "imperfekt minne".

II. Bakgrund

A. Stratego

Stratego är ett krigsbrädspel där två spelare försöker överlista varandra. Det finns många varianter på Stratego i vilka man har varierat spelplanens storlek, valörer på pjäser, antal pjäser eller rörelseregler. Ursprungliga Stratego spelas på ett 10x10 brädet med två "sjöar" som är fyra rutor stora. Vardera spelare har 40 spelpjäser med olika valörer och egenskaper. Båda spelarna börjar med samma sorts pjäser men får ställa upp dem hur de vill och pjäsernas valörer är dolda för motståndaren (se Fig. 1). Efter uppställningen gör spelarna turvis drag då de förflyttar sina pjäser. Alla pjäser förutom spejare (valör 2), flagga och bomber får förflytta sig till en intilliggande ledig ruta, det vill säga inte diagonal förflyttning och inte rutor som blockeras av egna pjäser eller sjöar. Spejaren kan förflytta sig fler än en ruta i samma riktning (likt torn i schack). Bomber och flaggan kan inte flyttas. Målet är att antigen fånga motståndarens flagga eller fånga alla motståndarens rörliga pjäser så att hen inte längre kan göra några drag. Man fångar pjäser genom att flytta en pjäs till en ruta som redan

upptas av en motståndarpjäs, båda pjäsernas valörer visas och, förutom i några special fall, pjäsen med lägst rank tas bort från spelbrädet. Om valörerna är lika tas båda bort från brädet. Specialfallen är:

- Minörer (3) är de enda som fångar bomber, i alla andra fall vinner bomben och den lämnas kvar.
- Spionen (1) vinner mot fältmarskalken (10) om spionen är den som anfaller.
- Alla rörliga pjäser kan fånga flaggan.

Fig. 1: Standard 10x10 Stratego uppställning [5].

B. Varianter av Stratego

Som nämnt tidigare finns det otaliga varianter av Stratego som på olika sätt ändrar på spelreglerna vilket i sin tur förändrar spelets karaktär och komplexitet. Följande är några vanliga ändringar som används i dagsläget.

1) Spelbräde: Den vanligaste variationen som görs är att ändra spelbrädets storlek. Oftast behåller man den kvadratiska formen men krymper spelbrädet till sju, fem eller tre rader och kolumner. I Fig. 2 illustreras 5x5-brädet. Vanligtvis finns inga sjöar på 3x3 planen. Det minsta spelbrädet, 3x3, är betydligt mindre än vanliga Stratego men erbjuder ändå en viss nivå av strategisk komplexitet, samt tillåter att spela flera snabba matcher.

De mindre spelbrädena har många regelvarianter på vilka pjäser som spelas med. Vilka man använder har stor påverkan på spelet, både strategimässigt och svårighetsmässigt. Att till exempel spela med spion, fältmarskalk och flagga i 3x3 är det annorlunda från spejare, fältmarskalk och flagga, då spionen kan fånga marskalken men det kommer spejaren aldrig kunna och spejaren kan röra sig på ett sätt som spionen inte kan. Exemplet är något trivialt men belyser de skillnader som kan uppstå.

2) Bara framåt, ingen reträtt: I denna regelvariant kan ens pjäser aldrig retirera utan kan endast röra sig framåt eller åt sidan. När en pjäs når motståndarens sida blir den fast där och kan då endast röra sig i sidleds längst bakersta raden. Denna regel leder till ett mycket mer aggressivt och snabbt spel samt en minskning i spelkomplexiteten då man i varje situation i regel har färre valmöjligheter.

3) Slumpmässig start: Pjäsupställningen väljs ut slumpmässigt från en homogen fördelning över alla möjliga pjäspermutationer. Med en slumpvald pjäsuppställning blir det enkelt sagt en sak mindre för spelaren att tänka på vilket reducerar spelkomplexiteten och speltiden.

4) Ledtrådar: I den ursprungliga regelboken för Stratego ska pjäser vändas tillbaka efter att de visats, vilket gör spelet mer minneskrävande för en mänsklig spelare. Alternativet är att låta pjäser förbli vända med valören synlig för motståndaren. En fortsättning på detta är att även markera pjäser som rört på sig även om dess valör inte har avslöjats ännu. Båda dessa ledtrådar är vanliga i online-versioner av Stratego.

Fig. 2: 5x5 Stratego under match [6].

C. Imperfekt information i Stratego

Spel som Stratego som bygger på att dölja information från sin motståndare faller inom kategorin av spel med *imperfekt information*. Dessa definieras formellt som spel i vilka det finns flera speltillstånd som för spelarna inte går att skilja åt [7]. Den dolda informationen ökar drastiskt spelets komplexitet både för människor och datorer då varje fiendepjäs skulle kunna vara en av flera olika valörer. Algoritmer som effektivt kan ta sig an spel med imperfekt information är få till antalet och svåra att implementera [8]. Stratego kan alltså ses som en blandning mellan schack och poker, då man både måste manövrera pjäser och även taktiskt bluffa motståndaren. Att bluffa är inte enkelt men helt centralt för att bli en bra spelare. En spelare som aldrig bluffar eller följer enkla bluffprinciper blir snabbt förutsägbar vilket kan utnyttjas av motståndaren. En spelare som bluffar väl kan dock vilseleda sin motståndare på alla möjliga vis. Till exempel kan man hota att anfalla med okända pjäser och få dessa att framstå som starkare än de egentligen är. Detta kan vara bra för att se hur motståndaren reagerar och utifrån detta dra slutsatser. Liknande kan man locka motståndaren med till synes svagapjäser och gillra fällor eller helt enkelt distrahera motståndaren från andra delar av spelbrädet.

D. Spelteori & notation

Beslutsfattande med imperfekt information kan modelleras generellt med hjälp av spel på omfattande form (eng: extensive games). Denna typ av spel representeras som spelträd med noder som speltillstånd och förgreningar som möjliga drag för spelaren på tur. Varje löv (sluttillstånd) ger poäng till spelarna

enligt spelreglerna, exempelvis vinst = 1, förlust = -1 och oavgjort = 0.

Centralt för spel med imperfekt information är begreppet *informationsmängder*, vilka bestäms av den information om speltillståndet en given spelare har tillgång till. Varje informationsmängd är en delmängd av speltillstånd som spelaren ifråga ej kan särskilja. I, t.ex, poker vet man inte motståndarens kort och kan därmed inte skilja på speltillståndet då motståndaren har par i ess eller par i fyror.

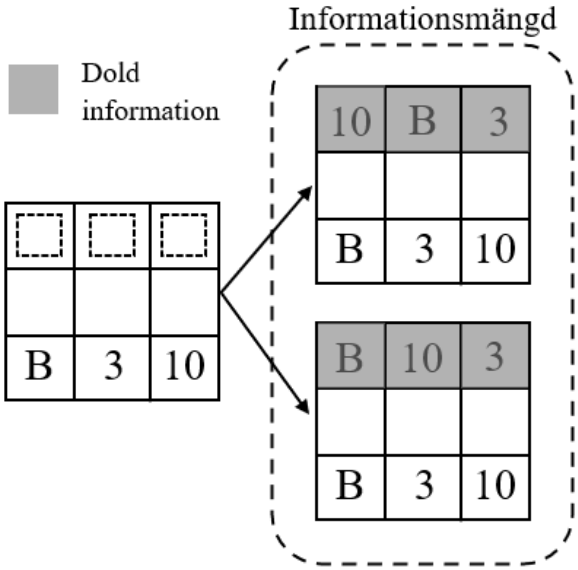


Fig. 3: 3x3 exempel på hur, från en spelares perspektiv, olika speltillstånd tillhör samma informationsmängd.

Fig. 3 illustrerar hur olika speltillstånd grupperas i en och samma informationsmängd från en spelares perspektiv. I figuren visas hur det ursprungliga spelbrädet (där det är övre spelarens tur att göra sin uppställning) förgrenar sig till olika spelbräden beroende på vilken uppställning den väljer. Från den nedre spelarens perspektiv tillhör dock dessa olika spelbräden en och samma informationsmängd eftersom den inte kan se valörerna.

Dessa egenskaper definierar vi nu formellt.

Definition [9, p. 200] *ett ändligt spel på omfattande form med imperfekt information har följande delar:*

- En ändlig mängd N med samtliga spelare.
- En ändlig mängd H med sekvenser av drag, där sekvenserna är alla möjliga spelhistoriker. Givet en sekvens $h \in H$ har vi att alla dess prefix, betecknade $h' \sqsubseteq h$, också tillhör H. Låt $Z \subseteq H$ vara mängden fullständiga historiker, alltså de historiker som ej är prefix till någon annan historik. $A(h) = \{a : ha \in H\}$ är de möjliga dragen från en ofullständig historik $h \in H \setminus Z$.
- En spelarfunktion P som mappar ofullständiga historiker $h \in H \setminus Z$ till spelaren $i \in N \cup \{c\}$ vars tur det är, där $P(h) = c$ innebär att slumpen styr nästa drag i spelet.
- En funktion f_c som mappar historiker där det är slumpens tur (h så att $P(h) = c$) till sannolikhetsfördelingar $f_c(a|h)$ över dragen $a \in A(h)$.
- För varje spelare $i \in N$ en partition $\mathcal{I}_i$ av historiker $\{h \in H : P(h) = i\}$ till informationsmängder I_i, med kravet att $A(h) = A(h')$ om h och h' tillhör samma medlem I_i i partitionen. För en informationsmängd $I_i \in \mathcal{I}_i$ låter vi $A(I_i) = A(h)$ för något $h \in I_i$
- För varje spelare $i \in N$ en poängfunktion $u_i(z)$ från Z till $\mathbb{R}$. Om $N = \{1, 2\}$ och $u_1 = -u_2$ så har vi ett *nollsummaspel på omfattande form*.

Förutom spelnotation inför vi även begrepp relaterade till spelarna. En strategi för spelare i, σ_i, är en sannolikhetsfördelning över de möjliga dragen $A(I_i)$ för varje informationsmängd $I_i \in \mathcal{I}_i$. Exempelvis, för en sten-sax-påse-spelare som alltid väljer sten skulle strategin σ_i kunna representeras av vektorn $[1, 0, 0]$. Låt även $\sum_i$ vara mängden av alla strategier för spelare i, σ_{-i} vara alla motståndares strategier och $\sigma = \{\sigma_i : \forall i \in N\}$ vara sammansättningen av alla spelarnas strategier. σ kallas strategiprofilen.

Värt att betona är att spel med imperfekt information kräver stokastiska strategier för att kunna spelas optimalt [9, p. 33]. Spel som luffarschack med perfekt information kan spelas optimalt med en deterministisk strategi: lägg alltid kryss i mittrutan. Spelar man deterministiskt i spel med imperfekt information avslöjar man ofta information för motspelaren som kan utnyttja det. En sten-sax-påse-novis som spelar deterministiskt genom att alltid välja sten kommer troligen förlora till en motspelare som märker det efter några matcher och börjar välja påse. Det har visats att den optimala strategin i sten-sax-påse är att med en tredjedels sannolikhet välja något av de tre dragen, alltså $\sigma = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$.

Vi definierar slutligen $\pi^\sigma(h)$ som sannolikheten att historiken h nås, d.v.s. sannolikheten att sekvensen av drag i h spelas, givet att spelarna använder strategierna i σ. Detta kan delas upp $\pi^\sigma(h) = \pi_i^\sigma(h)\pi_{-i}^\sigma(h)$ till bidraget från spelare i samt bidraget från alla motståndare (inklusive slump).

E. Exploitability

Exploitability är ett mått på en strategis styrka, eller rättare sagt svaghet, och är ett av det vanligaste måtten för utvärdering inom strategiframtagning.

Givet en statergiprofil σ definieras *best response* (för en given spelare) som den strategi som maximerar den förväntade poängen för spelaren om alla motståndare använder σ. Värdet, *best response value*, för denna strategi när den spelas av spelare *i* ges av $b_i(\sigma_{-i}) = \max_{\sigma_i' \in \sum_i} u_i([\sigma_i', \sigma_{-i}])$ då $u_i(\sigma) = \sum_{h \in Z} u_i(h)\pi^\sigma(h)$. Exploitability mäter hur mycket en strategi avviker från den teoretiskt optimala och i ett zero-sum spel med två spelare ges detta utav $\epsilon_\sigma = b_1(\sigma_2) + b_2(\sigma_1)$ [10].

F. Counterfactual Regret Minimization, en överblick

Counterfactual regret minimization (CFR) är ett sätt att utvärdera olika drag i en given situation (informationsmängd) och framställa en optimal strategi. CFR utför det genom att testa alla möjliga drag och sedan beräkna "regrets" som representerar hur bra eller dåligt ett visst drag visade sig vara denna gång. Regrets beräknas genom att se vilka slutsresultat man kan uppnå efter att ha gjort ett visst drag. Exempelvis, om ett drag garanterar att du förlorar matchen så anses det

draget dåligt och dragets regrets återspeglar detta genom att sättas negativa. På motsvarande sätt kommer dragets regrets bli positiva om det överlag leder till goda slutresultat. När alla situationer (informationsmängder) utvärderats kan en strategi utformas med hjälp av dessa regrets. I ett spel som Schack med perfekt information skulle det räcka att göra detta endast en iteration (betydligt enklare sagt en gjort) för att hitta den optimala strategin. Detta beror delvis på att varje informationsmängd motsvarar exakt ett speltillstånd. Som vi vet är detta inte fallet i spel med imperfekt information där en informationsmängd kan motsvara flera olika speltillstånd. Det gör att när CFR går igenom spelträdet kommer flera olika speltillstånd mappas till samma informationsmängd vilket leder till att CFR besöker samma informationsmängd flera gånger i en trädtraversering (iteration). Att beräkna regrets till en informationsmängd blir därmed svårare än tidigare eftersom utfallet från en viss handling inte längre är entydigt definierat (ex. om draget är att anfalla en fiendepjäs på en viss ruta blir resultatet av den handlingen olika beroende på vilken valör den pjäsen visar sig ha). Lösningen till detta är att göra en viktad summa av regrets från de olika speltillstånden i varje informationsmängd. Denna summa, kallad "regret sum", blir det som används för att beräkna strategin.

Väldigt viktigt distinktion att göra är att denna strategi inte är den spelteoretiskt optimala strategin vi söker. Den spelteoretiskt optimala strategin fås genom att ta medelvärdet av alla iterationers strategier när antalet iterationer går mot oändligheten. En liknelse kan göras till mänsklig erfarenhet. Vi lär oss inte endast från den senast spelade matchen utan har förmodligen utvecklat vår spelstil med ackumulerad kunskap från många matcher. Även det vi gjorde tidigt när vi ännu inte blivit särskilt duktiga bidrog till att utöka erfarenheten och göra oss bättre. Det finns olika varianter på CFR algoritmer vars syfte är att kunna konvergera till en stark strategi snabbare genom att fokusera på specifika delar av spelet och inte hela tiden utvärdera alla möjliga situationer. Vi kommer fokusera främst på det så kallade stokastiska (eller Monte-Carlo) variationerna [10] som beskrivs i nästa stycke.

G. Stokastiska CFR algoritmer

Stokastiska CFR algoritmer, främst kända som Monte-Carlo CFR (MCCFR), används oftast för spel vars spelträd är för stora för en full trädtraversering som i grundalgoritmen (vanligt kallad Vanilla CFR). Dessa algoritmer bygger på att i olika situationer göra slumpmässiga urval (eng: sampling) av handlingar istället för att utforska alla möjligheter. På så sätt begränsar man sig till en mindre del av spelträdet i varje iteration och sänker beräkningskostnaden samt tidsåtgång per iteration. Trots att man i varje iteration endast traverserar en begränsad del utav trädet bör man, efter tillräckligt många iterationer, nå alla speltillstånd. Tanken är att den minskade beräkningskostnaden per iteration ska vara tillräcklig för att balansera ut den även minskade konvergensen per iteration och leda till att algoritmen når jämvikt snabbare [10]. Spelträdet består av olika noder där varje nod representerar ett visst speltillstånd. Dessa noder, tillstånd, kan i sin tur delas in i två grövre kategorier, slumpnoder (eng: chance nodes) och beslutsnoder (eng: decision nodes). Befinner man sig vid en beslutsnod betyder detta att någon spelare kan fritt välja mellan ett eller flera drag. En slumpnod å andra sidan innebär att något av flera utfall slumpmässigt kommer ske [11]. Ett bra exempel är spelet Monopol där man rullar tärningen för att bestämma hur många steg man ska ta (en slumpnod) och därefter får fritt besluta hurvida man ska köpa den gata man landat på (beslutsnod). Vanligtvis består spel utav en blandning av slump- och beslutsnoder precis som Monopol men vissa spel som Schack saknar slumpnoder. Standard Stratego har precis som Schack inga slumpnoder, men varianter som den med slumpstart inför denna typ nod i spelträdet. Det finns många variationer på MCCFR-algoritmer [10], vi kommer fokusera på två av dessa som använder chance sampling och external sampling.

1) Chance sampling: Chance sampling bygger på att utföra de slumpmässiga händelseurvalen vid spelets slumpnoder. Detta innebär att när algoritmen (Chance CFR) når en slumpnod kommer den slumpmässigt välja ut en av de möjliga utfallen istället för att utforska alla [11]. I vårt Monopolexempel skulle algoritmen, istället för att utforska händelseförloppen för alla möjliga tärningskast (2, 3, ..., 12), välja ut ett av dessa, exempelvis 3. Resterande utfall förväntas utvärderas i de kommande iterationerna. Fullständig algoritm beskrivs i [11].

2) External sampling: External sampling är en utökning av chance sampling där man även slumpväljer vissa beslutsnoder. En algoritm med external sampling (External CFR) gör händelseurval vid alla noder förutom beslutsnoderna tillhörandes en vald spelare. Algoritmen körs alternerande med spelare ett eller två som vald spelare [10]. I Monopol skulle man göra urval vid alla tärningskast och när motspelaren kan köpa gator men utforska alla alternativ när den valda spelaren kan köpa gator. Fullständig algoritm beskrivs i [12].

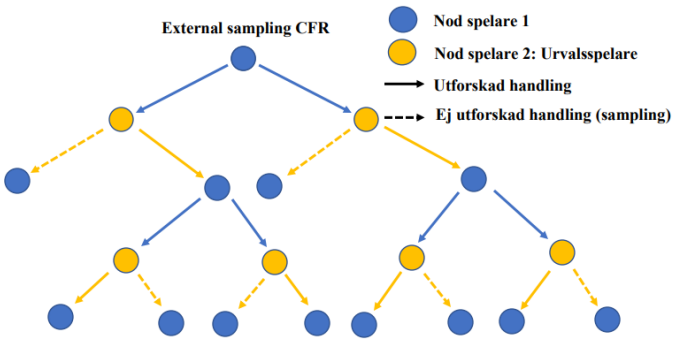


Fig. 4: Iteration för External CFR.

I Fig. 4 illustreras en exempeliteration för External CFR i ett påhittat spelträd. Spelträdet har noder tillhörandes de två spelare samt två möjliga handlingar per nod som leder spelträdet vidare. I figuren är spelare 1 en vanlig CFR-spelare som utforskar alla möjliga drag medans spelare 2 är den valda urvalsspelaren som väljer ett av de två möjliga dragen att utforska. I figuren visas dessa outforskade drag som sträckade pilar. Noderna dessa leder till kommer inte utforskas och följaktligen kommer inte heller några dotternoder till dessa utforskas. Att dessa grenar skärs bort resulterar i att betydligt

färre noder besöks vilket innebär en lägre beräkningskostnad per iteration.

H. Trimning

Trimning (eng: pruning) är ett tillägg som kan implementeras i CFR för att drastiskt minska beräkningskostander per iteration. Istället för att hela tiden utvärdera exakt alla speltillstånd väljer man bort alla de tillstånd som aldrig kommer att nås. Detta görs genom att i en given beslutsnod för en av spelarna utvärdera sannolikheten att motspelaren når den noden. Om sannolikheten är noll kan man sluta utförska denna gren [10].

I. Imperfekt minne

Idéen med imperfekt minne är att minska antalet informationsmängder genom att glömma delar av spelhistoriken. En spelare med perfekt minne kommer ihåg all information som tidigare avslöjats för denne genom spelet gång, och kommer även ihåg den exakta ordningen som den fick informationen. Spelare med imperfekt minne uppfyller detta vilket innebär att de glömmer viss information och kan därmed inte särskilja vissa spelhistoriker som en spelare med perfekt minne skulle kunna särskilja [13]. Från algoritmens perspektiv förminskas spelet och blir därför mindre krävande att utvärdera. Ur ett praktiskt implementationsperspektiv finns huvudsakligen två olika tillvägagångsätt gällande imperfekt minne. Det första tillvägagångsättet är att endast modifiera mappningen mellan spelhistorikerna och informationsmängderna. Det betyder att man fortfarande traverserar spelträdet som tidigare men att flera spelhistoriker grupperas samman till en informationsmängd. Det andra alternativet är att utöver modifieringen i mappningen även ändra hur CFR traverserar spelträdet. Eftersom samma informationsmängd kommer besökas flera gånger under en iteration så genomförs en del upprepade beräkningar. Detta kan optimeras genom att endast utföra dem en gång och vikta resultatet. Väldigt viktigt för båda varianterna är att alla spelhistoriker som grupperas har samma tillåtna handlingsalternativ. Är detta inte är uppfyllt ger spelardrag inte unika speltillstånd.

Fördelen med den första implementationen är att den är väldigt enkel och kräver inga större förändringar av CFR. Det är även mindre minneskrävande än perfekt minne eftersom de finns färre informationsmängder. Fördelen med den andra versionen är att den inte bara är mindre minneskrävande utan även mindre beräkningskrävande. Nackdelen är att implementationen är betydligt svårare. I båda fallen sker dock detta på bekostnad av spelstyrkan.

J. Utmaningar

De strategier CFR beräknar sparas tabulärt efter varje speltillstånd. Detta är nödvändigt för att uppnå den teoretiskt optimala strategin och lämpar sig bra för små simpla spel såsom mini-versioner av poker. För stora spel med stora antal speltillstånd medför tabulering dock problem. Stratego med sina 80 pjäser på ett 10x10 rutnät har otroligt många speltillstånd, och för varje speltillstånd måste optimala strategin beräknas och lagras. Även de mindre varianterna kan bli stora utmaningar för den begränsade beräkningskraften och minnet av en personlig hemdator.

En annan utmaning är att få forskare har arbetat med CFR, vilket har lett till låg variation i litteraturen. Den litteratur som finns tillgänglig innehåller svårtolkade matematiska definitioner ofta utan intuition eller förklaringar. Många detaljer som nämns lämnas ofta outforskade eller med bristande förklaringar för hur de påverkar algoritmerna. Även de exempel som tas upp i artiklarna är centrerade kring specifika spel och det generella fallet ges sällan explicit vilket gör det svårt att veta vad som är spelspecifikt och vad som är allmänt.

III. Metod

A. Implementation

Implementationen gjordes i Python 3.9 och som stöd användes C++ implementationen av Kenshi Abe på GitHub [14]. De olika algoritmerna som implementerats är Vanilla CFR, trimmad CFR, Chance CFR, External CFR och Vanilla CFR med imperfekt minne. Stratego implementerades på en generell form som tillåter snabba regel-, spelbrädes- och pjäsändringar. Med stöd av Trenners artikel om CFR [15] gjordes även en implementation av exploitability.

B. Experiment 1: Algoritmjämförelse

Detta experiment syftar till att studera konvergenshastigheten för de implementerade CFR-varianterna med perfekt minne. För detta ska konvergenskurvorna för Vanilla CFR, trimmad CFR, External CFR och Chance CFR jämföras. Konvergenskurvorna mäter exploitability som en funktion av totalt nådda noder. Totalt nådda noder är summan av antalet besökta informationsmängder över alla iterationer, och används som ett implementationsneutralt mått på beräkningstid. Ren mätning av beräkningstid varierar mellan datorer, operativsystem och programmeringsspråk vilket inte lämpar sig för jämförelser.

Vi använder ett 3x3 spelbräde med en bomb, en fältmarshalk och en minör. Dessutom används reglerna “Bara framåt, ingen reträtt”, “Slumpmässig start” samt ledtrådarna gällande rörelse och avslöjad valör.

C. Experiment 2: Spelstyrka och minne

Målet med detta experiment är att avgöra hur spelstyrkan förändras med olika mängder spelminne samt studera sambandet mellan strategistorlek (antal informationsmängder) och spelstyrka. Spelare med perfekt minne kommer ihåg alla drag sen början av en match, medan en spelare med fyra i spelminne endast kommer ihåg de senaste fyra dragen. Alla möjliga minnesnedskärningar, även den med noll spelminne, ser fortfarande spelplanen från sitt perspektiv och kommer ihåg om fiendepjäser har rört sig eller avslöjats under tidigare strid. Alla spelare i experimentet kommer tränas med trimmad CFR. Efter träning analyseras minnets påverkan på konvergensen med hjälp av konvergenskurvor. För att studera sambandet mellan startegistorlek och spelstyrka jämförs den uppnådda spelstyrkan med totala startegistorleken för de olika minnesmängderna. Samma spelbräde och regler som i Experiment 1 används.

IV. Resultat

A. Experiment 1: Algoritmjämförelse

I Fig. 5, Fig. 6 och Fig. 7 visas konvergenskurvorna för Vanilla CFR, trimmad CFR, External CFR samt Chance CFR. Fig. 6 visar en närbild på intervallet $[0, 5 \cdot 10^6]$ från Fig. 5 medan Fig. 7 är en när närbild på på intervallet $[10^7, 5 \cdot 10^7]$.

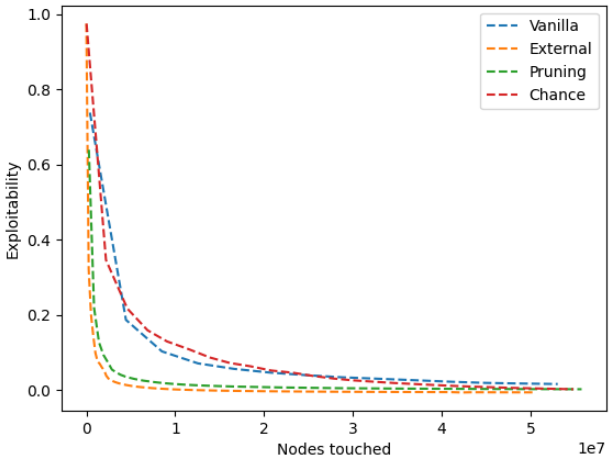


Fig. 5: Konvergenskruvor för algoritmvarianterna där exploitability anges som funktion av nådda noder.

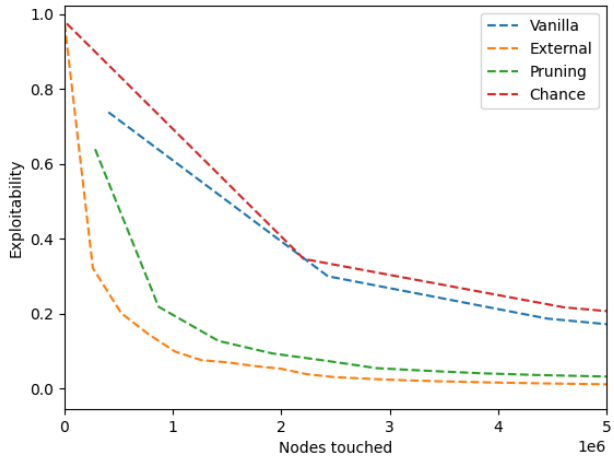


Fig. 6: Närbild på intervallet $[0, 5 \cdot 10^6]$ av konvergenskurvorna från Fig. 5.

Fig. 5 visar att all strategier oberoende av algoritm konvergerar mot den spelteoretiskt optimala strategin med exploitability 0 (givet tillräckligt många iterationer). Dessa konvergenskurvor bekräftar att algoritmerna framställer strategier vars egenskaper stämmer överens med det som stipulerats i teorin. Skillnaderna i startvärde beror på att exploitability beräknas för första gången efter en iterationen. Närbilderna på konvergenskurvorna i Fig. 6 och Fig. 7 visar tydligare hur konvergenstiden skiljer sig åt mellan algoritmerna. Vi ser att External CFR samt trimmad CFR har en signifikant brantare kurva jämfört med Vanilla CFR och konvergerar cirka 5 respektive 10 gånger snabbare. Vi kan även se att Chance CFR inte uppnår en snabbare konvergens än Vanilla CFR i de tidiga stadierna av strategiframtagningen, som visas i Fig. 6. I ett senare stadie av framtagningen, som illustreras i Fig. 7, kan vi dock se att Chance CFR presterar något bättre.

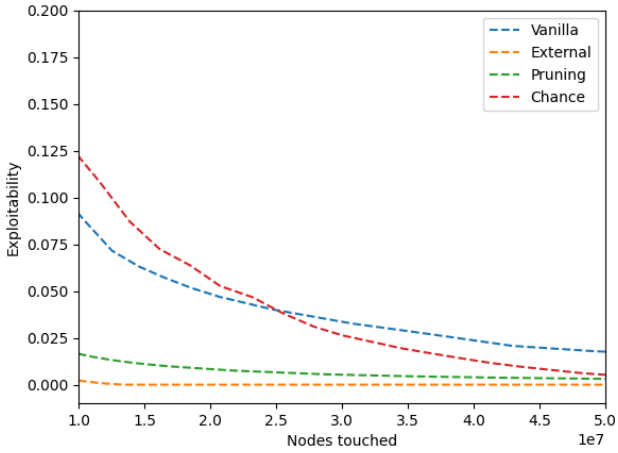


Fig. 7: Närbild på intervallet $[10^7, 5 \cdot 10^7]$ av konvergenskurvorna från Fig. 5.

B. Experiment 2: Spelstyrka och minne

I Fig. 8 och Fig. 9 visas träningen för Vanilla CFR med olika mycket spelminne. Båda dessa visar närbilder av konvergenskurvorna, Fig. 8 på intervallen $[0, 3 \cdot 10^6]$ och Fig. 9 på intervallet $[10^7, 3 \cdot 10^7]$.

I grafen ser vi att algoritmerna körda med lägre spelminne konvergerar något snabbare. Detta är att förvänta eftersom att det mindre spelminnet reducerar antalet olika informationsmängder och gör att varje informationsmängd besöks och uppdateras flera gånger under en CFR-iteration. I Fig. 9 ser vi dock svagheten med lägre spelminne: algoritmen med minne 0 konvergerar inte till 0 exploitability. Detta har den intuitiva förklaringen att en motståndares drag ger ledtrådar om pjäsernas valör, och att glömma de ledtrådarna kan bara försämra din spelstyrka. Intressant att anmärka är att fullt minne ej syns i dessa figurer eftersom den är “under” minne 6 som har nästa identiska värden.

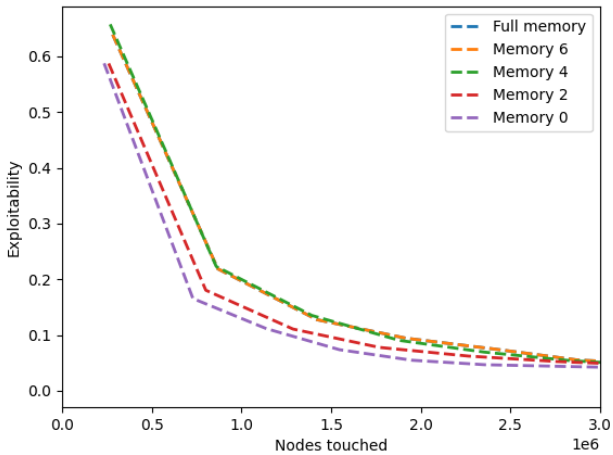


Fig. 8: Närbild på intervallet $[0, 3 \cdot 10^6]$ av konvergenskurvorna för Vanilla CFR med olika minnemängder.

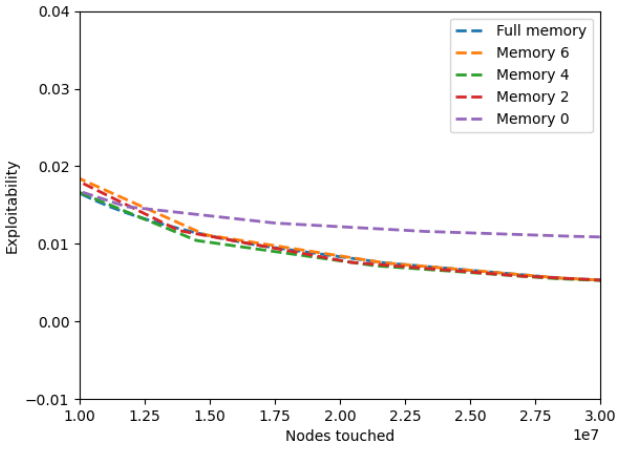


Fig. 9: Närbild på intervallet $[10^7, 3 \cdot 10^7]$ av konvergenskurvorna för Vanilla CFR med olika minnemängder.

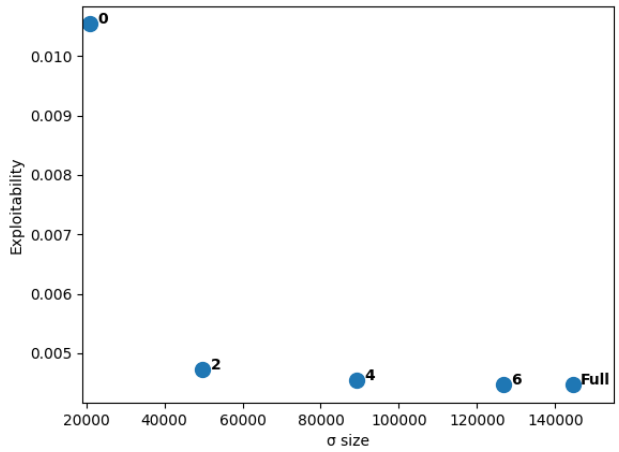


Fig. 10: Slutgiltig exploitability för algoritmerna mot storleken på deras strategiprofiler. Minnesbegränsningarna för varje algoritm anges vid sin datapunkt.

Uppoffringen i spelstyrka ger dock betalt i minskat lagringsutrymme som krävs för CFR. Fig. 10 illustrerar de olika minnesgränsernas slutgiltiga exploitability mot antalet informationsmängder i strategiprofilen σ. Vi ser att fullt minne och två drag i minne skiljer sig väldigt lite i exploitability men ungefär med en faktor tre i strategistorlek. Att inte komma ihåg några tidigare drag alls visar sig ge betydligt större förlust i spelstyrka än de andra minnesgränserna. Detta säger oss att motståndarens senaste drag är mycket viktigare strategiskt än de tidigare dragen.

V. Slutsatser

I detta arbete har vi studerat startegiframtagning med CFR-algoritmer i imperfekt informationsspelet Stratego. Vidare har vi analyserat hur dessa CFR-varianter och tillägget imperfekt minne påverkar konvergenstid respektive minnesanvändning.

Resultaten från våra experiment har visat att Monte-Carlo-varianten External CFR är avsevärt snabbare än grundalgoritmen. Även trimning visar sig vara en tydlig förbättring. Monte-Carlo-varianten Chance CFR visade sig ge en mindre signifikant förändring i konvergens jämfört med trimmad CFR och External CFR. Detta beror sannolikt på att slumpnoderna i spelträdet utgör en väldigt liten andel av de totala noderna just för Stratego. Därmed beter sig Chance CFR precis som Vanilla CFR i majoriteten av speltillstånd.

Experiment 2 visade att relativt stora minnesbegränsningar kan göras utan större förluster i spelstyrka. I nedskalade Stratego verkar den viktigaste informationen vara spelbrädets nuvarande situation med information om pjäser som visast och rört på sig samt de två senaste dragen som gjorts. Med denna information är spelstyrkan i stort sätt motsvarande den som åstadkoms med perfekt minne. Resultaten visade även att det finns betydelsefulla minnesfördelar med att begränsa spelminnet. Användingen av imperfekt minne blir således en avvägning mellan implementationens ändamål och tillgängliga datorresurser.

VI. Framtida forskning

Större och mer komplexa spel, som fullskaliga Stratego, har för stora spelträd för att en agent på rimlig tid ska kunna tränas med CFR. För framtida arbeten skulle vi vilja vidare utforska användningen av funktionsapproximation, exempelvis neurala nätverk, i kombination med CFR.

Vi skulle även vilja undersöka komplexiteten av den nedskalade versionen av Stratego som har använts. Detta för att säkerställa att reduktionen inte har gjort spelet trivialt.

VII. Författarnas tack

Vi vill tacka våra handledare Mika Cohen och Farzad Kamrani för deras stöd och vägledning genom arbetet.

Referenser

[1] (2018, Dec.) Alphazero: Shedding new light on chess, shogi, and go. [Online]. Available: https://deepmind.com/blog/article/alphazero-shedding-new-light-grand-games-chess-shogi-and-go

[2] T. Allen, *The evolution of wargaming: From chessboard to marine doom in War and Games*, 1st ed. Republic of San Marino: The Boydell Press, 2002, p. 231–250.

[3] N. Brown, A. Bakhtin, A. Lerer, and Q. Gong, "Combining deep reinforcement learning and search for imperfect-information games," *arXiv preprint arXiv:2007.13544*, 2020.

[4] C. Resnick. (2018, Juli) Deepstack. [Online]. Available: https://www.depthfirstlearning.com/2018/DeepStack

[5] (2021, April) Barndomsminnen. [Online]. Available: https://www.pinterest.se/pin/116389971594926095/

[6] (2021, April) Strategy pro. [Online]. Available: https://play.google.com/store/apps/details?id=com.famlinkup.stratego

[7] N. A. Risk and D. Szafron, "Using counterfactual regret minimization to create competitive multiplayer poker agents," *Proceedings of the Ninth International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Maj 2010.

[8] J. R. S. Blair, D. Mutchler, and C. Liu, "Games with imperfect information," *Working Notes AAAI Fall Symposium on Games: Planning and Learning*, 1993.

[9] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*. Cambridge: MIT Press, 1994.

[10] M. Lanctot, K. Waugh, M. Zinkevich, and M. Bowling, "Monte carlo sampling for regret minimization in extensive games," *Advances in Neural Information Processing Systems*, vol. 22, 2009.

[11] T. Neller and M. Lanctot, "An introduction to counterfactual regret minimization," *Proceedings of Model AI Assignments, The Fourth Symposium on Educational Advances in Artificial Intelligence (EAAI-2013)*, vol. 3, Juli 2013.

[12] R. Gibson, “Regret minimization in games and the development of champion multiplayer computer poker-playing agents,” *Ph.D. Dissertation, University of Alberta*, 2014.
[13] M. Lanctot, R. Gibson, N. Burch, M. Zinkevich, and M. Bowling, “No-regret learning in extensive-form games with imperfect recall,” *Proceedings of the International Conference on Machine Learning (ICML)*, 2012.
[14] K. Abe “bakanaouji”. (2020, Jan.) cpp-cfr. [Online]. Available: https://github.com/bakanaouji/cpp-cfr
[15] T. Trenner. (2020, Maj) Building a poker ai part 7: Exploitability, multiplayer cfr and 3-player kuhn poker. [Online]. Available: https://ai.plainenglish.io/building-a-poker-ai-part-7-exploitability-multiplayer-cfr-and-3-player-kuhn-poker-25f313bf83cf

Bluffing AI in Strategy Board Game

Albin Henriksson and Johan Leijonhufvud

***Abstract*—Games have been a field of interest for research in artificial intelligence for decades. As of now, it is over 5 years ago an AI for the strategy game Go, AlphaGo, beat world champion Lee Sedol 4-1, which was considered to be an enormous milestone for AI. Our goal is to make an AI that can play the classic strategy board game Stratego at a competent level. This is achieved by making the AI learn by repeatedly playing against itself to figure out what strategy to use in various situations by using CFR - counterfactual regret minimization. According to our experiments, we were able to accomplish our goal in making a Stratego AI that could play at a sophisticated level for a smaller version of the game. We concluded that it was able to play better than an amateur human player.**

***Sammanfattning*—Spel har varit ett intresseområde inom utvecklingen av artificiell intelligens i årtionden. Det är redan fem år sedan AlphaGo slog världsmästaren Lee Sedol i Go 2016, vilket betraktas vara ett stort steg för utvecklingen av AI. Vårt mål är att skapa en AI som kan spela strategispelet Stratego på en kompetent nivå. Detta kommer att implementeras genom att AI:n spelar mot sig själv en stor mängd gånger och uppdaterar sin strategi baserat på konceptet CFR - counterfactual regret minimization. Enligt våra experiment lyckades vi med vårt mål i att skapa en kompetent Stratego AI för en mindre version av Stratego. Vår uppfattning är att den spelar bättre än en människa på amatörnivå.**

***Index Terms*—Strategy game, Stratego bot, Tactical AI, Counterfactual regret minimization, Chance sampling.**

Supervisors: Mika Cohen and Farzad Kamrani

TRITA number: TRITA-EECS-EX-2021:196

I. Introduction

Games have always been an important field of interest in the research and development of artificial intelligence. They can be used as a metric to test and evaluate AI. In 2016, AlphaGo beat the reigning world champion Lee Sedol at the strategy game Go [1]. This was considered a huge milestone in the field of AI as some experts thought this would not be possible for another decade. In particular, this report is going to focus on the strategy board game Stratego, which is an imperfect information game. Imperfect information games differ from perfect information games in that they are significantly more difficult to create efficient algorithms for. It is impossible to know what the oppoonent's best move is when you have do not have information of where their pieces are located. For this reason, imperfect information games are often scaled down to be solvable for a computer. In this report, Stratego is studied for two different board sizes. We will analyze how the size and complexity of the game board affects runtime and performance. The goal is to make the stratego bot play at a relatively high level. This will be discussed more in-depth later.

A. Research topic

One reason to study games is in order to evaluate theoretical strategies in practise. For this Stratego bot in particular, we will use a Counterfactual regret minimization (CFR) algorithm to train our AI. CFR has been used to solve other games, such as poker. Other algorithms have been used in Stratego before, for example variations of the minimax algorithm [2]. The theory behind CFR will be explained later in the Background section.

B. Stratego ruleset

Traditionally, the game is played on a 10x10 board where each player place their 4x10 pieces on their side of the board and the two middle rows are left empty. An example of this can be seen in figure 1. In this game of Stratego, each player has one flag, one spy denoted by S, a few bombs and the rest of the pieces possess different ranks where 1 is the best and 9 is the worst. The other player's pieces are hidden, making this an imperfect information game which will be discussed more later. All movable pieces (i.e. every piece apart from the flag and bomb) can take one vertical or horizontal step every round except for the Scout (the piece with a rank of 8) that can move any number of steps as long as it is in a vertical or horizontal line. No piece can move onto the two 2x2 ponds in the middle though. Whenever a piece moves onto a piece on the opposing team, a confrontation between them occurs and whichever piece has the highest rank wins the showdown. The losing player does not have to reveal what rank their piece possesses, only the winning piece gets revealed. If a movable piece moves onto a piece that turns out to be a bomb, both pieces are destroyed. There are two ways of winning a traditional game of Stratego. Either by capturing the opponents flag, or by eliminating all opposing pieces. There are many different strategies one can exploit in Stratego. One such strategy is to place all your strongest pieces in the front to prevent your opponent from penetrating your defense. Another strategy is to surround the flag with bombs and strong pieces to make it difficult for the opposition to capture it. As in many other strategy games, it is often beneficial to try and trick the opponent by bluffing or making unexpected moves, Stratego is no different in this regard. For example, the opponent is likely to expect your flag to be well protected, meaning they would expect your flag to be in an area with a high concentration of highly ranked pieces. Therefore, one can make a cluster of highly ranked pieces without placing the flag there to fool the opponent. The rules are gather from [3].

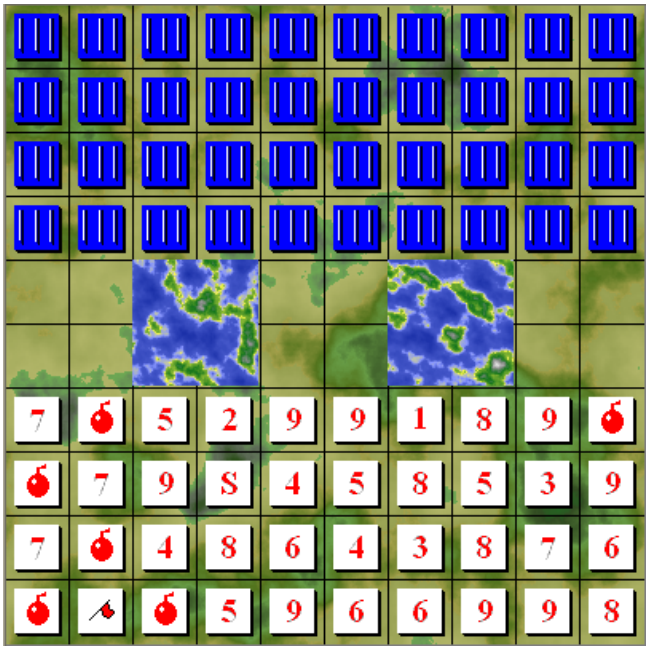


Figure 1. A computer version of 10x10 Stratego [4]

II. Background

A. Imperfect information, Nash equilibrium and mixed strategies

There are two types of games, perfect information games, and imperfect information games. A perfect information game is where both players have access to all information available. Two such games are chess and Go, where the entire board is visible to both players. Poker is an example of an imperfect information game since each player sees only their own hand. The strategies in imperfect information games rely heavily on probability theory to make moves. In Stratego, each player can only see their own pieces, making it an imperfect information game. One of the key differences between AI's for perfect information games and imperfect information games is that it is much more difficult to make efficient algorithms for imperfect information games such as full scale Poker than it is for perfect information games such as Chess or Go. For this reason, smaller downscaled versions of imperfect information games are often used instead when building AI bots. The AI develops its strategies by playing against itself. After having played many times against itself, the strategy the AI uses doesn't change much. This brings us to the concept of a Nash equilibrium. A Nash equilibrium is a game theoretical state where both players have no incentive to alter their strategy. This is often referred to as a solution in two player strategy games because the individual player does not gain anything by deviating from their strategy [5]. For example, in rock paper scissors, the Nash equilibrium is to play each action one third of the time. When working with imperfect information games the best strategy is often a mixed strategy, meaning there are two or more actions that will be made with some probability. [6]. The Nash equilibrium in rock paper scissors uses a mixed strategy .A mixed strategy can be used in order to bluff, or to prevent the opponent from exploiting your strategy.

B. Counterfactual Regret Minimization (CFR)

We need to cover CFR, as it is a core concept in this project and will be used in the implementation of the Stratego AI. Much of the theory has been gathered from [7] but modified. The idea behind CFR is to calculate the difference between the utility u a certain action yields, and compare it with the utlity for every other action. By doing this, one can put a numerical value on how much one regrets a certain action. We have below used similar notatations as is used in [8]. Let us introduce some notation.

- S is the set of all possible information sets. An information set for a player is any information that they have access to and which could have an affect on the outcome of the game. In all the cases below let I denote any information set in S, let I_i denote an information set for player i and let I_{-i} denote an information set for the opponent.
- $A(I)$ is the set of all legal actions at information set I. The number of legal actions given information I is therefore $|A(I)|$.
- Let $\vec{\sigma}^t(I) = (p_1^t(I), ..., p_{|A(I)|}^t(k))$ be the strategy vector at information set I and iteration t. Where $p_j^t(k)$ represents the probability of action a_j.
- For any $a \in A(I)$ then $u(I \rightarrow a)$ is the utility at information set I for any legal action a. The utilities are given in terminal information sets, meaning the game is already finished but we have to check who has won or if it is a draw. This is often done by using: $u(I \rightarrow a) = 1$ if a results in a win, $u(I \rightarrow a) = 0$ if a is a draw and $u(I \rightarrow a) = -1$ if a is a loss. The utilities at information sets that are non terminal are given by the formula

$$u(I) = (u(I \rightarrow a_1), ..., u(I \rightarrow a_{|A(I)|})) \cdot \vec{\sigma}^t(I). \quad (1)$$

  If action a on information set I leads to the opponent's information set I_{-i}, then $u(I \rightarrow a) = -u(I_{-i})$. The goal is to maximize $u(I)$ at every information set.
- Let $\pi_i^t(I)$ be the probability of reaching information set I at iteration t when we count the actions for the other player (not player i) as having probability 1 to reach information set I.

With $A(I) = \{a_1, ..., a_{|A(I)|}\}$, we define the regret vector representing every legal action as

$$\vec{r}^t(I) = (u(I \rightarrow a_1) - u(I), ..., u(I \rightarrow a_{|A(I)|}) - u(I)) \quad (2)$$

The components j of the regret sum vector $\vec{R}^T(I_i)$ for some information set I_i are defined as

$$R_j^T(I) = \begin{cases} \sum_{t=1}^T \pi_{-i}^t(I) r_j^t(I) & \text{if } 0 < \sum_{t=1}^T \pi_{-i}^t(I) r_j^t(I) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

for some iteration T. Note that $\vec{R}^T(I) \geq \vec{0}$. In the first iteration, we do not have any previous iterations to calculate the regret from, so $\vec{r}^0(I) = \vec{R}^0(I) = \vec{0}$. Since we are

regarding probabilities and vectors, it is important to define the probability vector.

Definition 1. [9] Let $\vec{v}$ be a vector of length L. If

$$\sum_{j=1}^{L} v_j = 1, \tag{4}$$

where v_j is element j in $\vec{v}$, then $\vec{v}$ is a probability vector.

The strategy vector will be updated from iteration $T \rightarrow T+1$ in the following manner:

$$\vec{\sigma}^{T+1}(I) = \frac{\vec{R^T}(I)}{\sum_{j=1}^{|A(I)|} R_j^T(I)} \tag{5}$$

unless

$$\vec{R}^T(I) = \vec{0}. \tag{6}$$

In this case, the strategy will be uniformly distributed. From equation 5, it is clear that $\vec{\sigma}^{T+1}(I)$ is a probability vector as it should.

Now, the weighted strategy vector is calculated as

$$\vec{S^T}(I) = \sum_{t=1}^{T} \pi_i^t(I)\vec{\sigma}^t(I) \tag{7}$$

where

$$\vec{S^0}(I_i) = \vec{0}. \tag{8}$$

With the weighted strategy vector $\vec{S^T}(I)$, we also consider the probability of reaching information set I in our strategy. Given a final iteration F, the final strategy is given by

$$\vec{\sigma}^F(I) = \frac{\vec{S^F}(I)}{\sum_{j=1}^{|A(I)|} S_j^F(I)}, \tag{9}$$

unless

$$\vec{S^F}(I) = \vec{0}. \tag{10}$$

In this case, the strategy will be uniformly distributed. Note that $\vec{\sigma}^F(I)$ is still a probability vector. The Stratego CFR bot employs a strategy to maximize the utility at each information set. In practise, this is done by repeatedly playing against itself and re-evaluating its strategy at each iteration and updating the strategy until it converges to a Nash equilibrium. As we will see below, rock paper scissors converges to a Nash equilibrium that consists of choosing each action one third of the time.

C. CFR for rock paper scissors

To get some intuition for CFR and how it can be applied in a simple game, let's consider two games of rock paper scissors. First, note that all available actions for both players at every information set is $A = \{rock, paper, scissors\}$. For simplicity, a win has a payoff of 1, a tie has a payoff of 0 and a loss has a payoff of -1. Before the first game $\vec{R}^0(I) = \vec{0}$ for every information set $I \in S$. Recall that an information set is the information a given player currently has access to. In the case of rock, paper scissors, the information a player knows at the beginning is nothing, let's denote this by $s = \varnothing$. At the first iteration the strategies are uniformly distributed: $\vec{\sigma}^0(I) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. Let's calculate the regret for player 1 when player 1 goes for rock (completely randomly). We begin by denoting the relevant information sets below:

- $s = \varnothing$
- $k = s \rightarrow \{\text{player 1 chooses rock}\}$
- $p = s \rightarrow \{\text{player 1 chooses paper}\}$
- $f = s \rightarrow \{\text{player 1 chooses scissors}\}$
- $k_1 = \{\text{player 1 chooses rock, player 2 chooses rock}\}$
- $k_2 = \{\text{player 1 chooses rock, player 2 chooses paper}\}$
- $k_3 = \{\text{player 1 chooses rock, player 2 chooses scissors}\}$

We have

$$u(k) = \frac{1}{3} \cdot (u(k_1) + u(k_2) + u(k_3)) = \frac{1}{3} \cdot (0 + (-1) + 1) = 0. \tag{11}$$

$$u(s) = \frac{1}{3} \cdot (u(k) + u(p) + u(f)) = 0 \tag{12}$$

. In this case the regret sum vector of player 1 is given by: $\vec{R}^1(s) = (u(k) - u(s), u(p) - u(s), u(f) - u(s)) = (0, 0, 0)$. The strategy becomes uniform $\vec{\sigma}^1(s) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, according to equation 6. In this case, the CFR algorithm converges to Nash equilibrium after one iteration because it started at a uniform strategy, which is optimal in rock paper scissors. If we would have started with any other strategy than $\vec{\sigma}^0 = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ it still would have converged to the uniform strategy after a sufficient amount of iterations. This is an example of the basic idea of CFR which will be implemented in our Stratego AI.

D. Bluffing Computer

An interesting thing about CFR bots like the one used in Stratego is that they can learn bluffing strategies. One such example is in the game of Kuhn poker, which is a simplified version of poker devised by american mathematician and game theorist Harold W. Kuhn. In two player Kuhn poker, the deck consists of three cards, namely a king, queen and jack. Each player draws one card and player 1 chooses to bet one coin or check. If player 1 bets, player 2 can either call or fold. If player 2 calls, both players add one more coin to the pot and show their cards and the player with the most valuable card wins the pot. When Kuhn poker is played with CFR, one notices that the bot learns to bluff after a sufficient amount of iterations. The bot learns that pulling a jack without bluffing is always a loss since it's the weakest card, so it employs a bluffing strategy to make up for its otherwise low value. This can be seen in [10].

E. Problem statement

In this project, we have built a Stratego bot for two different game sizes. The goal was to construct a competent bot that can play equally good as a human, or ideally, better. Furthermore, we have examined how the game complexity and size affects runtime of the program and how we can decrease convergence time of the algorithm by implementing variations of CFR.

III. Method

A. Time complexity of CFR for Stratego

Assume a game board of size n, a maximum number of turns M, k movable pieces each, and I iterations for CFR to run. As we established in the section about Chance sampling with CFR, the number of starting positions is $(n!)^2$. This means that we call the CFR function $(n!)^2$ times at each iteration. At each player's turn, they can move their k movable pieces in at most 3 directions (forwards, left or right). This means that there are at most $3k$ moves a player can legally make on their turn. Given that there are at most M turns, the total number of games is at most $(n!)^2(3k)^M$. With CFR running over I iterations, the total time complexity results in

$$O_{CFR}(I(n!)^2(3k)^M). \tag{13}$$

Note that this truly is an upper bound. This model assumes 3 feasible moves for each piece every turn, which is not true in practise. On the first turn for each player, there are only k legal actions since they must move forwards. Furthermore, if one of the pieces is next to a wall or has been eliminated, the number of feasible moves is also lower than $3k$. Combining these facts results in less than $3k$ feasible moves per turn on average.

B. Implementation in Python

As discussed in section about CFR, we will use CFR to calculate regret of the bot, and use this to update the strategy at each iteration. Every information state I_i will be the information of where all pieces of player i are, what the values of those pieces are, which move it is and where your opponent's pieces are. This will make the bot play progressively more intelligently and, eventually, the strategy will reach a Nash equilibrium [**?**].

Definition 2. Just as in rock paper scissors, we will define a win of any one game as a payoff of 1, and a loss as a payoff of -1.

$$u(win) = 1 \tag{14}$$

$$u(loss) = -1. \tag{15}$$

C. Evaluation of the bot

In order to evaluate the proficiency of the bot, we will make a human player play against the computer 50 times and keep track of the win rate, and compare the results with when the computer plays against itself. A consequence of definition 2 is that the expected value is simply given by the difference of the win rate and loss rate. By referring to the probability of a win as $p_i(win)$ and a loss as $p_i(loss)$, we obtain

$$E_i = p_i(win) - p_i(loss). \tag{16}$$

In Stratego,

$$\begin{cases} p_1(win) = p_2(loss) \\ p_1(loss) = p_2(win). \end{cases} \tag{17}$$

This results in

$$E_1 + E_2 = 0. \tag{18}$$

Equation 16 will be combined with a corollary of the Central limit theorem from statistics.

Theorem 1. [11]

If $X_1, X_2, ..., X_n$ is a sequence of independent equally distributed stochastic variables with expected value μ, standard deviation $\sigma > 0$, and a large enough sample size n, then

$$\frac{\sum_{i=1}^{n} X_i}{n} \in N(\mu, \frac{\sigma}{\sqrt{n}}), \tag{19}$$

In other words, for a large enough sample size, we can approximate the probability distribution of the expected value as $N(\mu, \frac{\sigma}{\sqrt{n}})$. This will be useful to analyze the plausability of the model in section IV.

D. Chance sampling with CFR

Given a game board of size $n \times n$, the number of starting positions for any one player is $n!$, giving us the total number of possible starting positions as $(n!)^2$. This means that the runtime of the program increases significantly with game size, and we had to make some adjustments to the algorithm in order to keep the runtime down when playing larger games. Instead of running the CFR algorithm for all possible starting positions as in the 3×3 version of Stratego, the algorithm now randomly picks one starting position and updates the strategy vector based on the strategy it finds given the current information state. By doing this, we only consider one starting position at each iteration instead of $(n!)^2$. Naturally, this is going to decrease runtime at each iteration, but less information will be provided, and consequently, more iterations will be needed for convergence. Our goal is to decrease convergence time, and we will examine whether this change in the algorithm is justified given our objective.

E. 3×3 Stratego

Due to computational limitations, smaller versions than the typical 10×10 Stratego have been used in this report. This is in reality, not unheard of. For instance, the Stratego Pro app uses a 5×5 board. First, we will treat 3×3 Stratego using CFR, and expand to a 4×4 board later. It is important to note that the game is not trivialized on smaller boards. The 3×3 board consists of a grid where both players start off with a flag and two pieces with different ranks on their side of the board as in figure 2. Starting positions are randomized with the middle row being initially empty. Starting with player 1 and taking turns, each player moves their piece one step horizontally or one step forwards. We have restricted the pieces to be unable to move backwards. This is in fact an established variant of the board game. When both players have made 5 moves each without capturing the flag and both players can still move pieces, the game is declared a draw. If, on a given turn, a player can't move any of their pieces, they lose the game. We want to investigate whether player 1 or player 2 has an advantage.

Figure 2. An example of a possible starting state n 3×3 Stratego. F denotes the flag, 1 denotes the weaker movable piece and 2 denotes the stronger movable piece.

F. 4×4 *Stratego*

In our version of 4×4 Stratego, in addition to the pieces we had in 3×3 Stratego, both players also possess a moat piece. This piece is immovable just like the flag, and destroys any piece that moves onto it. An example of a game board can be seen in figure 3. Another key difference is that there

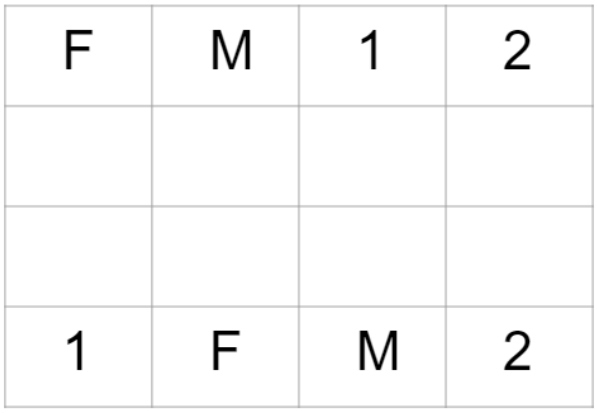


Figure 3. An example of a possible starting state in 4×4 Stratego. Apart from the pieces we had in 3×3 Stratego, we also have M - the moat piece.

are two middle lanes that are initially empty. This means that the number of possibilities on any one round is, on average, greater than in 3×3 Stratego since it will be possible to move in more directions. This, together with the increase in possible starting positions from $(3!)^2 = 36$ to $(4!)^2 = 576$, justifies usage of chance sampling to potentially decrease convergence time. As in 3×3 Stratego, it is also interesting to see if player 1 or player 2 has an advantage.

IV. Results and Discussion

A. *Computer vs Computer* (3×3 *Stratego*)

Looking at figure 4, we observe several interesting results. The algorithm converges relatively quickly, only after about 15 iterations, and that being the player who makes the first move is advantageous. Interestingly, the game never results in a tie, meaning the game always finishes in 10 rounds or less.

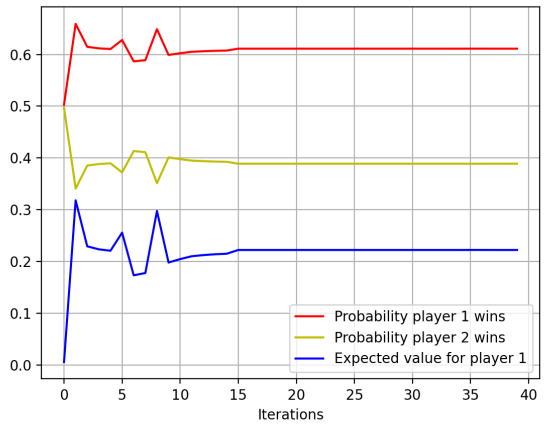


Figure 4. Both players use the CFR algorithm in 3×3 Stratego. Recall that the expected value for player 1 is $E_1 = p_1(win) - p_1(loss) = p_1(win) - p_2(win)$

B. *Human vs Computer* (3×3 *Stratego*)

We played against the CFR bot 50 times to see if the theoretical results in figure 4 match up with what we obtained in the experiment. At the computer's turn in the human vs computer experiment, the current game board is manually entered in Python so that the computer knows from which position it should evaluate the next move. It will then return the move it chooses and that move will be executed on the physical game board. The human will then make their move and the cycle continues. The results of this experiment are presented in table I.

By examining the results in figure 4, we see that it does

	Computer (player 1)	Human (player 2)
Wins	27	23
Ties	0	0
Win rate	0.54	0.46

Table I
Results of Human vs CFR on a 3×3 board

not quite match up with the results we obtained in the human vs computer experiment since the win rate of each player is not quite the same. However, this result is not statistically improbable. By using equation 16, we obtain $E_{1,exp} = 0.54 - 0.46 = 0.08$. The definition of the standard deviation yields $\sigma = 0.997$. Now using theorem 1, we obtain that the expected value of player 1 for $n = 50$ follows

$$E_1 \in N(E_{1,exp}, \frac{\sigma}{\sqrt{n}}) = N(0.08, 0.141) \tag{20}$$

We are interested in how likely the expected value for computer vs computer, $E_{1,theo}$ from figure 4 is, given $E_{1,exp}$. We want to solve the value of x that satisfies

$$E_{1,exp} + x\frac{\sigma}{\sqrt{n}} = E_{1,theo}. \tag{21}$$

Solving this yields

$$x = \frac{\sqrt{n}(E_{1,theo} - E_{1,exp})}{\sigma} = \frac{\sqrt{50}(0.22 - 0.08)}{0.997} = 0.993. \tag{22}$$

where $E_{1,theo} = 0.21 \pm 0.01$ from figure 4. We used $E_{1,theo} = 0.21 + 0.01 = 0.22$ to get an upper limit. This means that $E_{1,theo}$ is at most 0.993 standard deviations larger than the experimental value $E_{1,exp}$. Given that

$$P(\mu - 0.993\sigma < X_i < \mu + 0.993\sigma) = 68\% \tag{23}$$

for a stochastic variable X_i that follows a normal distribution, the probability that the theoretical value satisfies

$$|E_{1,exp} - E_{1,theo}| > 0.993\sigma \tag{24}$$

is at least 32%, i.e, not particularly unlikely. This means $E_{1,theo}$ is not unlikely given our experimental result. Despite the threshold of 10 rounds per game, we did not play more than 7 rounds in any of the 50 games. Another thing to note is that the strategy the computer uses does not care how many rounds it takes to win, as long as it wins at some point. At a few occasions, the games could have been shorter, but since at the current game state, it did not matter whether the computer moved forward or sideways, it chose arbitrarily and it ended up winning one round later than neccesary. It is also interesting that the computer chose some counterintuitive moves. For example, in one instance the computer used a mixed strategy of going forwards 80% of the time, but in that particular game, it chose the 20% move of going to the right. This was an example of a bluff, the obvious choice would have been to move forwards.

C. Computer vs random strategy (3×3 Stratego)

We measure how the trained CFR bot performs against a random strategy where all the opponent's moves are equally likely. The result of this experiment is shown in figure 5. Now, the win rate converges faster than before and ends up at approximately 0.87. The reason the win rate of player 1 does not go higher than this is because of the randomness in the game. Player 2 can get lucky and win.

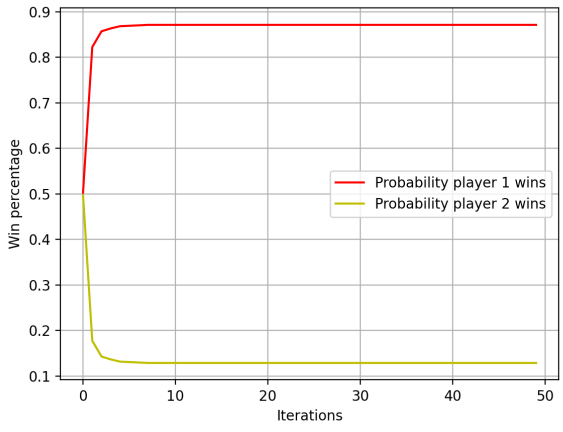


Figure 5. Only player 1 improves with CFR in 3×3 Stratego. Player 2 uses a uniformly distributed strategy.

D. Computer vs Computer (4×4 Stratego)

Just as in 3×3 stratego both players play every possible starting position once at each iteration. This means that every iteration in figure 6 represents playing $4! \cdot 4! = 576$ starting positions. As we can see in figure 6, being player 1, i.e. the one making the first move, is a slight disadvantage due to the negative expected value $E_{1,theo}$. Our result in 3×3 Stratego was the contrary.

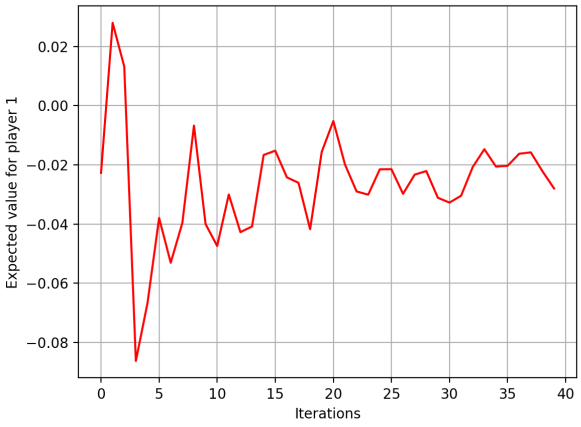


Figure 6. Both players use the CFR algorithm for 4×4 stratego.

E. Human vs Computer (4×4 Stratego)

As we did in 3×3 Stratego, we also tested the strategy against a human player by playing 50 times. The results are presented in table II. This gives us an experimental expected value of $E_{1,exp} = 0.28 - 0.32 = -0.04$. This time, we obtain $\sigma = 0.774$. Using theorem 1, we would expect

	Computer (player 1)	Human (player 2)
Wins	14	16
Ties	20	20
Win rate	0.28	0.32

Table II
RESULTS OF HUMAN VS CFR ON A 4×4 BOARD

$$E_1 \in N(E_{1,exp}, \frac{\sigma}{\sqrt{n}}) = N(-0.04, 0.109) \tag{25}$$

Comparing this with the result in figure 6 which converges to $E_{1,theo} = -0.02 \pm 0.02$. This is within 0.4σ of $E_{1,exp}$. The probability that $E_{1,theo}$ satisfies

$$|E_{1,exp} - E_{1,theo}| > 0.4\sigma \tag{26}$$

is at least 69%. This means that the experimental results in 4×4 Stratego supports the assumption of $E_{1,theo}$ being true more than they did for 3×3 Stratego. Interestingly, in 4×4 Stratego, the games ended in ties almost half of the time. However, this is not too surprising upon analyzing further. Given that there is no clear way of knowing which one of the opponent's immovable pieces is the flag and which is the moat, the computer learns that on average, it is highly likely to

interact with the moat piece when attacking a piece that hasn't moved. This results in a stalling tactic where both players move back and forth in certain situations which ends in a tie after the 10th round. More mixed strategies are being used in 4×4 Stratego compared to 3×3. The reason for this is because the added row and column makes it easier to get away with bluffing, whereas in 3×3 Stratego, player 1 moves up its strongest piece on the first move most of the time.

F. Conventional CFR vs Chance sampling CFR (4×4 Stratego)

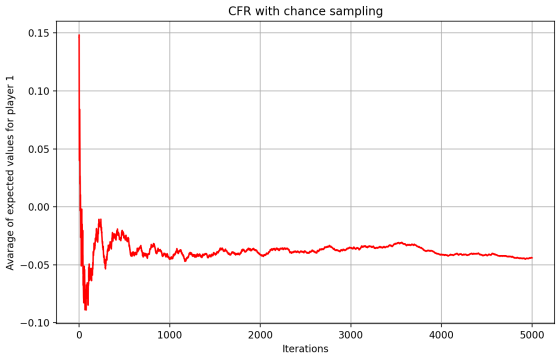


Figure 7. CFR with Chance Sampling in 4×4 Stratego

In figure 7, the CFR algorithm with chance sampling runs over 5000 iterations in contrast to figure 6 where only 40 iterations are done. The main difference being that now, only 1 starting position is considered at each iteration instead of all possible combinations. Thus, in the expression for the time complexity in equation 13, the factor $(n!)^2$ is removed. This results in a time complexity for chance sampling CFR of

$$O_{CS}(I(3k)^M). \tag{27}$$

The trade-off here being that I has to increase for convergence, as we saw in figure 7. The runtimes of the two are presented in table III

	Conventional CFR (figure 6)	Chance sampling CFR (figure 7)
Runtime	27h and 8 min	6h and 37 min

Table III
RUNTIME OF CONVENTIONAL CFR VS CHANCE SAMPLING CFR

Despite far more iterations when using chance sampling, the overall runtime is much lower. The convergence is also clearer compared to the conventional CFR algorithm. In table IV we let the conventional CFR algorithm play 4×4 Stratego against the chance sampling algorithm. It seems clear that despite a shorter runtime for chance sampling CFR, it still outperforms the the conventional CFR algorithm in 4×4 Stratego. As has been established before, in a game with two equally intelligent players, it is preferred to start as player 2. Despite this, chance sampling still beats conventional CFR when starting as player 1.

	Conventional CFR (figure 6)	Chance sampling CFR (figure 7)
Expected value as player 1	-0.0966	0.0398
Expected value as player 2	-0.0398	0.0966

Table IV
EXPECTED VALUE OF CONVENTIONAL CFR VS CHANCE SAMPLING CFR IN 4×4 STRATEGO

G. Player 1 vs Player 2

It has been made clear that playing as player 1 in 3×3 Stratego is advantageous, while being player 2 is beneficial in 4×4 Stratego. The reason it is beneficial to play as player 1 in 3×3 Stratego is because it is typically risk free to move towards player 2's row unless their high ranked piece is in the same column. Typically, it is the lower ranked piece and the flag that did not move on player 2's first turn, meaning that player 1's higher ranked piece will either find the flag, or eliminate the lower ranked piece of player 2 if it moves forwards twice and player 2's high ranked piece is not in the same column. This means there's a high probability to find the flag after 2 moves, or eliminate player 2's low ranked piece. Either way, it is a good position. In 4×4 Stratego, the main difference is that the introduction of the moat piece changes the strategy completely. It is no longer risk free to confront one of your opponents pieces that have not moved, since there is a good chance that is the moat piece. It is not as clear why player 2 has an advantage in 4×4 Stratego, but it is likely because player 1 reveals more information about their pieces sooner, thus giving player 2 an informational advantage.

V. CONCLUSION

In summary, we concluded that we achieved our goal of making the Stratego bot play at a competent level. The results showed that playing Stratego is very costly computationally, even for smaller boards. The difficulty being that Stratego is an imperfect information game that requires more computational power than perfect information games such as Chess. Running a CFR algorithm for a complex imperfect information game like Stratego takes a long time with increased game complexity. We implemented the algorithm in Python which is time efficient for writing code but not for running the completed program. In order to speed up the runtime, one could implement the algorithm in a programming language that has high computational speed such as C, or C++. One of the most exciting and unexpected results were that the computer chose moves that, at first glance did not make sense. At several occasions while playing against the bot for 3×3 and 4×4 Stratego, it was unclear if there were bugs in the program or if we had missed to consider all possible moves. Upon analyzing the moves further, it was clear that the computer had considered outcomes we had not and used bluffing strategies. This further convinced us of the intellectual prowess of AI and made it clear how it can be used to gain strategic insights that can be applied in the real world.

ACKNOWLEDGMENT

The authors would like to thank Mika Cohen and Farzad Kamrani for providing us with guidance and plenty of useful advice along the way.

REFERENCES

[1] C. Lee, M. Wang, S. Yen, T. Wei, I. Wu, P. Chou, C. Chou, M. Wang, and T. Yan, "Human vs. computer go: Review and prospect [discussion forum]," *IEEE Computational Intelligence Magazine*, 2016.

[2] K. Stengård, "Utveckling av minimax-baserad agent för strategispelet stratego," Bachelor's Thesis, Lund University, Sweden, 2006.

[3] (2021) The classic game of battlefield strategy. [Online]. Available: https://www.fgbradleys.com/rules/Stratego.pdf

[4] A. Kaufmann. (2004) Screenshot of stratego for zillions of games engine. [Online]. Available: https://upload.wikimedia.org/wikipedia/commons/0/05/Stratego.png

[5] A. Kajii and S. Morris. (1997) The robustness of equilibria to incomplete information. [Online]. Available: https://www.jstor.org/stable/2171737?seq=9#metadata_info_tab_contents

[6] M. Walker and J. Wooders. (2016) Mixed strategy equilibrium. [Online]. Available: https://link.springer.com/referenceworkentry/10.1057%2F978-1-349-95121-5_2277-1

[7] R. Gibson, N. Burch, M. Lanctot, and D. Szafron. (2021) Efficient monte carlo counterfactual regret minimization in games with many player actions. [Online]. Available: https://papers.nips.cc/paper/2012/file/3df1d4b96d8976ff5986393e8767f5b2-Paper.pdf

[8] M. Lanctot and T. W. Neller. (2013, Jul.) An introduction to counterfactual regret minimization. [Online]. Available: http://modelai.gettysburg.edu/2013/cfr/cfr.pdf

[9] L. Sadun, *Applied Linear Algebra: The Decoupling Principle*. USA: American Mathematical Society, 2008.

[10] J. Sermeno. (2021) Vanilla counterfactual regret minimization for engineers. [Online]. Available: https://justinsermeno.com/posts/cfr/

[11] G. Blom, *Sannolikhetsteori och statistikteori med tillämpningar*. Olsztyn, Poland: Studentlitteratur, 2008.

CONTEXT P – PART II
ARTIFICIAL INTELLIGENCE

POPULAR DESCRIPTION

Terminator: the new kind of medical assistant

The process of figuring out what is wrong with a human body today is a quite long and tedious process, there is often a shortage of doctors that specializes in certain fields. As a patient, all you want to know is what is wrong with you, and how to fix it. In such a sensitive and critical field as healthcare, an extensive waiting time could mean the difference between life and death.

To avoid these long waiting times, the hospital processes must become more efficient. For example, within diagnostics of oral diseases in Stockholm, dentists have to wait for an assessment from a Malmö-based pathologist to be able to move forward with their patient care. To ease the load on the dentists they could use the latest discoveries within Artificial Intelligence (AI).

To be able to diagnose a patient it takes humans years of training and experience. It is therefore very time consuming and hard to acquire intuition and a so-called "trained eye". This is where AI can really outperform humans, as a diagnosis of an AI only takes days to train. When trained, an AI could evaluate a patient within minutes and do this without making biased decisions based on human needs and emotions.

Hold up; if an AI could be trusted, would one not simply take medical advice from the Terminator? If you go to the hospital, it could be hard to trust that a computer is able to diagnose you correctly. Is the purpose of using AI in the medical sector to replace doctors, or is it rather to assist them by doing the heavy lifting and providing support to their decisions? Today's research and development is focused on creating supportive tools for medical use. We have a long way to go before society can even consider replacing doctors, if ever. Until then the healthcare sector remains in dire need of trained professionals.

SUMMARY OF PROJECT RESULTS

The need for time and cost effective, scalable solutions within our global healthcare system is extensive. These needs are mainly driven by strong trends within urbanization, globalization, and an increasing population growth. These problems are even more critical and pronounced within the field of Pathology. Firstly, the driving factors mentioned above create an increasing demand on pathology services. Secondly, pathologists are rare. For example, there are no oral pathologists in Stockholm, meaning Karolinska Institutet (KI) currently must send images of patient's tissue samples across the country to receive a diagnosis. Thirdly, in terms of reproducibility and reliability, pathology is a complex field with subjective elements, meaning experienced pathologists sometimes draw conflicting conclusions on the same data.

The latest and most promising solution to address these needs is built on digitized patient data, so called digital pathology. This is done by implementing computer-aided diagnostics (CAD) systems to aid the pathologists with their diagnoses. For these purposes machine learning (ML) models could be utilized, hereafter referred to as "AI methods". AI is software that teaches itself to perform certain tasks. These tasks are based on given labelled data from which a model first learns to detect and classify classes in the dataset, for example cell types. Later in the evaluation stage, the model can find and predict classes on an unseen but similar dataset.

Furthermore, the global healthcare system is lacking standardized experimental procedures, which are essential for these AI methods to work. For example, two tissue samples from different labs using the same sample-process can look very different. Hence, before applying these AI methods, normalisation of the data could lead to improved results.

In digital pathology, the use of AI methods can assist the workflow of pathologists, making it more efficient, less time consuming and less subjective. AI methods can also be a central part in training new pathologists as it can easily provide learning material, making the training process more effective and less biased.

The project groups in P3 analysed the use of three colour normalisation algorithms, so called *filters*, and ensembled the results to boost the performance of a deep neural network model (an AI method) that finds and classifies cells in oral tissue samples. The network in question is named EfficientNet. A previously established pipeline was used on the supercomputer Alvis, the pipeline was adapted to accept specific input-images. The groups focused on the *Khan, Reinhard, Histogram* and *Macenko* filters within the open-source library Warwick Toolbox. Training on filtered images showed a slight improvement in performance in comparison to the unfiltered dataset.

The work done by the project groups in P3 opens up the possibility for further work in similar areas. One is to extend the experiment by training on other types of datasets, such as different tissue samples or using other colour normalisation algorithms. Another interesting area is the effects of image compression in image processing. Some filters used showed extreme colour deviations compared to the expected output of the filter. By looking into the consequences of the use of compressed images in colour normalisation, a better understanding could be gained on how images should be pre-processed.

IMPACT ON SOCIETY AND ENVIRONMENT

Assessing the impact AI has on society and the environment in general is a substantial task. We shall begin, therefore, by stating the scope of our analysis: “Digital Pathology with AI tools for diagnostics”, hereafter referred to as “DPAI”. Furthermore, in this analysis one must differentiate between fully-automated (total pathologist replacement), clinically supportive and educational DPAI technologies. We will first address how DPAI has an impact on both societal and individual levels, and later we will discuss the sustainability and environmental aspects.

Firstly, one has to appreciate that in terms of impact on society versus impact on individuals, the concepts could be considered equivalent. Pathologists are specialists in the healthcare system and support dentists when diagnosing patients, especially in rare and difficult situations and diseases. If DPAI could help these professionals to make their diagnoses, it would lead to more efficient health care, where more people could get diagnosed at a possibly lower cost and shorter time. It could be argued that the increased benefit for the individual equally benefits society, as the number of individuals affected is a significant portion of the population.

One of the goals of using DPAI is to obtain objective results that do not discriminate based on any criteria. Since the AI reflects the results of the labelled dataset it is trained on, we need to be very conscious of the choice of datasets. Therefore, it is crucial that the selection of the dataset is thoroughly assessed so that as few biases as possible are passed on to the AI.

When using a DPAI to help in important decisions with large consequences, such as diagnosing patients, it is important to think about where the responsibility lies and how mistakes should be handled. There are many people involved in the process of making a diagnosis that therefore could be considered partially accountable: those who develop the DPAI, those who decide which images should be used for the training dataset, and perhaps most importantly the pathologist who makes the decision to use the results from the DPAI. Finally, one could argue that there should always be a specialist responsible for and making the final decision in processes involving human lives, but there should in any case be a legal framework put into place.

Regarding environmental reflections, one could consider the impact to be negligible. However, one part of the DPAI development involves a large energy consumption which could have an environmental impact at a large scale. To create (“train”) an accurate model within the field of DPAI, one usually needs a model with over hundreds of millions of parameters. To speed up the training process one usually uses a specific energy-intensive hardware, such as a High-Performance Computer Cluster of GPUs. However, when one has trained a model, the energy consumption when using it is insignificant. The energy cost used in training the model could be seen as an investment, where one must consider the trade-offs for every specific application and situation. At a societal level, this fact should be considered and taken into account, but should not prevent the continued development of DPAI.

With increasing need for new tissue samples to be used in training DPAIs, the privacy of the data used must be protected and possible leaks carefully protected. Tissue is extracted from patients, the samples are sliced, dyed, and then scanned. The resulting data is then used to train AI models. As data must be openly shared among the project members, which handle the models, it is of great importance that the data used cannot be traced back to the patients to avoid potential invasion of privacy. To do this, data must be anonymised before used in any AI processing or training.

Another ethical dilemma when using DPAI is deciding when they are “good enough” to use. What level of precision should be demanded of the tools to be used? Ideally, they would be 100% accurate all the time, but that is simply not possible. So where do we draw the line? Do the lines differ between supporting tools, fully automated tools and tools used for teaching medical students? Who decides where the lines are drawn and who are responsible for the possible consequences? It is important that these questions are considered when moving forward. The answers may come with time and experience, but for now we have to think carefully before we act.

Lastly, the inner workings of AI can often be described as a “black box”, meaning that it gets an input and produces an output, but as a human it is very hard, or sometimes even impossible, to understand the reasoning of an AI. Therefore, the trust a user has in DPAI often lies solely in its performance.

In conclusion, we see that DPAI has a low ecological impact, while the potential social and ethical benefits and consequences are substantial. Therefore, only the social and ethical effects should be carefully cultivated and balanced when implementing DPAI in healthcare.

Boosting CNN Performance in Digital Pathology Using Colour Normalisation and Ensembling

Emelie Kvarnström and Axel Tibbling

Abstract—**Researchers within digital pathology are endeavouring to develop machine-learning tools to support dentists when making a diagnosis. The purpose of this study was to investigate how applying colour normalisation (CN) algorithms on an oral, histopathological dataset would impact both machine-learning models and ensembles of models when classifying cell types.**

The dataset was run through four different CN algorithms by using a stain normalisation toolbox. The now five datasets (1 + 4) were then fed separately into a pipeline to create machine-learning models, specifically convolutional neural networks with EfficientNet architecture. Two different ensembles were studied, one that used all the models and one that used the three models with the highest test accuracy. Each model gave a cell type prediction of each cell. The ensembles super positioned their models' predictions of the same cell and used the results as their own predictions.

The models based on datasets created by two of the CN algorithms had a weighted, average accuracy of ca. four percentage points higher than the model based on the unnormalised dataset. Unexpectedly, the models based on the colour-normalised datasets had a *larger* standard deviation than the model based on the unnormalised dataset. All the models were generally bad at classifying two of the four cell types. Both the ensembles had a weighted, average accuracy of ca. ten percentage points higher than the model based on the unnormalised dataset, as well as a larger standard deviation. The increase in accuracy is significant and could move forward the timeline for when machine-leaning tools can be implemented into dentists' and pathologists' workflow.

Sammanfattning—**Forskare inom digital patologi strävar efter att utveckla maskininlärnings-verktyg som stödjer tandläkare när de ställer diagnoser. Syftet med denna studie är att utreda hur tillämpning av färgnormaliserande algoritmer (CN algoritmer) på ett oralt, histopatologiskt dataset påverkar hur både maskininlärningsmodeller och ensembler av modeller klassificerar celltyper.**

Datasetet kördes igenom fyra olika CN algoritmer med hjälp av en färgnormaliserings-verktygslåda. De nu fem dataseten (1 + 4) matades separat in i en "pipeline" för att skapa maskininlärningsmodeller, specifikt djupa neurala nätverk med EfficientNet arkitektur. Två olika ensembler skapades, en som använde alla modeller och en som endast använde de tre som hade högst noggrannhet på testsettet. Varje modell uppskattade celltypen för varje cell. Ensemblerna superpositionerade deras modellers uppskattningar för varje cell och använde resultaten som sina egna uppskattningar.

Modellerna som tränats på två av de färgnormaliserade dataseten ökade i viktad, snitt-noggrannhet med fyra procentenheter i förhållande till modeller tränade på det ursprungliga datasetet. Förvånansvärt nog så ökade även standardavvikelsen hos modeller tränade på de färgnormaliserade dataseten. Alla modeller var generellt dåliga på att klassificera två av de fyra celltyperna. Ensemblen uppnådde en viktad snitt-noggrannhet på ca. tio procentenheter mer än modeller tränade på det ursprungliga datasetet. Noggrannhetens signifikanta ökning kan leda till en tidigare implementering av maskininlärnings-verktyg i tandläkares och patologers arbetsflöde.

Index Terms—**Colour normalisation, Histogram, Khan, Macenko, Reinhard, ensemble, digital pathology, histopathology, deep learning, EfficientNet.**

Supervisors: Karl Meinke (KTH) and Rachael Sugars (KI).
TRITA number: TRITA-EECS-EX-2021:197

I. Introduction

In pathology, tissue samples are examined to make a diagnosis. In order to study small tissue samples, the samples are cut into thin slices. The parts of the tissue samples that are of interest, such as cellular components, are transparent. The samples are therefore dyed, often with the dyes hematoxylin and eosin (H&E), to make them visible as described by Feldman et al. [1]. The process of dyeing tissue samples is called *staining*. Ideally, staining the samples leaves them pink and purple, with clear contrasts. Many different factors affect the results, some of which are listed by Wick [2], Lyon and Horobin [3], e.g. how the samples were sliced, the thickness of the slices, whether the slices are damaged, different dye producers etc. Other outcomes include samples being heavily stained, lightly stained, and stained with different kinds of dye. Niazi et al. [4] points out that these variations complicate the work of the pathologists, effectively slowing down the workflow.

Furthermore, Pallua et al. [5] indicated that a significant number of pathologist were due to retire in the coming decade, and that the demand for and on these specialists was increasing. There is therefore a dire need for the workflow to become more efficient. This is one of the aims of the research area called *digital pathology*. With the introduction of digital-imaging tools, multiple new possibilities for improving the workflow arises. Some of these include implementing machine-learning (ML) models as a tool for pathologists.

These algorithms, particularly convolutional neural networks (CNN), show promise when analysing histopathological samples, as shown by Pontalba et al. [6] and Estreen [7], making them highly suitable for the task. One prominent issue of ML algorithms is their potential lack of robustness, in other words their performance transfers well to new data. Pallua et al. [5] explains that the robustness depends on the amount of training data available, but large and standardised initiatives to create the necessary datasets are currently non-existent, only in small instances. It is therefore hard to train a CNN that will produce equally good results everywhere. One step on the way of standardisation is to use colour normalisation (CN) algorithms to normalise the colours of the images, as done in

Pontalba et al. [6]. Using a small number of different CN algorithms, say k number of CN algorithms, one can generate $k+1$ number of datasets, and by utilising these datasets, $k+1$ different models can be trained. These can be combined into an *ensemble* by using the results of each model.

In this project, which is part of an ongoing collaborative project between KTH and KI that studies the application of ML in oral biology, the performance of both models trained on colour normalised images and ensembles made up of those models, were compared to a model trained on unnormalised images, as done in Pontalba et al. [6]. The dataset used in this study is a subset of the dataset compiled by Tollemar et al. [8]. It consisted of images of *chronic graft versus host disease (GVHD)* infected oral mucosa with different degrees of severity, ranging from healthy to severely inflamed. The position of some of the cells and their cell type (*inflammatory*, *lymphocyte*, *fibroblast and endothelial*, and *epithelial*) had been labelled by KI.

Building up to this project, two essential things were done. Firstly, a CNN algorithm with EfficientNet architecture was created by Estreen [7] and set up in a pipeline on SNIC's computing cluster *Kebnekaise* [9]. Since then, the pipeline has been transferred to their computing cluster *Alvis* [10], at Chalmers. Secondly, the dataset used (Tollemar et al. [8]) was labelled by KI.

II. Literature survey

A. Pathology, histology and digital pathology

Pathology is the science of diseases. According to Pontalba et al. [6] and Leong and Zhuang [11], a pathologist makes a diagnosis based on the colour and structure of tissue. This is called the *morphology*, and it provides the basis for determining both the aggressiveness and the subsequent treatment plan. A pathologist traditionally studies tissue on glass slides under a light microscope to determine a diagnosis. The area of studying the structure, or *morphology*, of tissue is *histology*.

According to Pallua et al. [5], as examinations become more complex, and the importance of personalised treatment plans increase, the need for pathologists is growing. At the same time, the workforce is dwindling as a lot of pathologists are due to retire and there are few young pathologists to replace them. For example, Pallua et al. [5] stated that Austria in 2020 had 299 active pathologists, of which 58% should retire within 10 years. These pathologists need to be replaced, but only a few of Austria's internships are occupied, which highlights the fact that the pathologist profession needs to become more attractive to young people.

The basis of digital pathology is whole slide imaging (WSI), which scans histopathological slides and digitises them. With this technology, a plethora of opportunities arose. Firstly, the process of studying slides became more efficient as it was easier for the pathologists to work remotely and share images with colleagues when making a diagnosis. This has much improved the workflow at hospitals without resident pathologists as they no longer have to send tissue samples for an external diagnosis. Pallua et al. [5] raise the fact that WSI has multiple times shown to be equal to the light microscope from a basic utilisation perspective. As WSI digitises slides, pathologists are now able to study multiple slides at the same time, facilitating the daily work of the pathologist. As tissue samples are being digitised, large-scale collaborations within digital pathology is becoming more possible.

The existence of WSI challenges light microscopes in the pathologists' workflow as well as enables the application of digital tools on pathological images. With the digital storage of the images, CN and ML algorithms and computational tools could, as Niazi et al. [4, p. 1] wrote, "...extend the frontiers of the pathologist's view beyond a microscopic slide...". The workflow could be further improved by the usage of digital tools, as they could perform tasks such as cell nucleus counts, locating important histological markers etc. for a pathologist to study further. He or she could then approach the image slides more holistically, without having to examine all of the images in detail.

As studied by Farahani et al. [12], the functionality and capacity of WSI devices vary, e.g. the time required to scan a sample and the colour contents of the resulting images. An example of varying colour content can be found in figure 4 in Smith et al. [13]. As different institutes use different WSI devices, the exact same digitised slide could have different colours, affecting comparability.

B. Oral mucosa and staining

The dataset used in this study is composed of sections of *oral mucosa*, the outer layer of the tissue on the inside of the mouth. The *epithelial* layer is the outer surface, and the *lamina propria* is the connective tissue, as can be seen in Figure 1. In this study, the models are trying to classify *inflammatory, lymphocyte, fibroblast and endothelial* as well as *epithelial cells*.

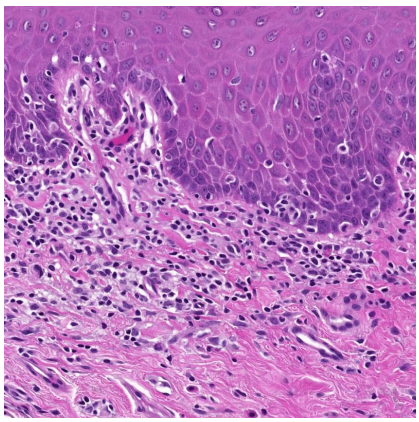


Fig. 1. Oral mucosal tissue showing epithelial cells in the upper layer and lamina propria underneath. The image is extracted from the dataset.

The process of getting extracted tissue onto a glass slide for a pathologist to study is extensive and essential. The steps of the process are explained in depth by Feldman et al. [1]. In short, it begins with dehydrating and fixating the tissue, often in formaldehyde, followed by a microtomy i.e. the tissue

is sliced into cell-thin sheets. The final step is colouring the tissue, *staining*.

Due to the nature of cells, tissue sliced cell thin is transparent and pathologists are therefore unable to study the morphology. To overcome this issue, the contrast is increased through staining, as explained by Alturkistani et al. [14]. The most widely used dyes are H&E. As Feldman et al. [1, p. 31] put it, "The hematoxylin and eosin (H&E) stained tissue section is the cornerstone of anatomical pathology diagnosis."

As the number of steps from extraction to staining is large, and the individual steps also demand accuracy, the results often present a large, unwanted variability. This is due to a large number of factors ranging from human error to the chemicals themselves, but there are key issues. Firstly, there is currently no standardised staining process worldwide, to achieve such a thing would be a immense task, as mentioned by Lyon and Horobin [3]. Secondly, the same kind of staining chemicals produced by different manufacturers yield different results. Bentaieb et al. [15] stressed that the visual appearance of the tissue greatly affects the quality and accuracy of the diagnosis. Decreasing the variability in the visual appearance is therefore essential.

C. Colour normalisation

Folmsbee et al. [16] showed that ML models trained on datasets with large variability perform poorly. This is often the case due to the variability in the staining process. To overcome this, a possible solution could be to run the images through CN algorithms.

To reduce the colour variation across a set of pathological images, an image with an ideal colour profile - a so-called *target image* - was selected. The colour profile of the target image, together with another image from the set, i.e. a *source image*, are then fed into a colour normalisation algorithm to produce a *normalised* version of the source image. The source image is normalised with the goal of making its colour profile more like that of the target image. There are multiple CN algorithms that do this, and some have been designed with a specific area of application in mind. Khan et al. [17] explain that in digital histopathological images, it is often the *stain vector* of the target, which consists of the absorption factors of the stain, that the algorithm endeavours to mimic in the source.

Four CN algorithms, utilised by Pontalba et al. [6], have been used in this study: Histogram, Khan, Macenko and Reinhard. The specific details of each algorithm lies outside the scope of this project. The reader is referred to the respective papers for a more in-depth explanation of the CN algorithms, but in short:

1) Histogram: *RGB Histogram specification* is the method of matching the histogram of the RGB channels of a source image to the RGB channels of a target image. This CN algorithm has been in use since 1986, when Lain [18] was published. Lain [18], Annadurai and Shanmugalakshmi [19] described in detail how an image's tonal distribution can be graphically represented with a histogram.

2) Khan: Created by Khan et al. [17] in 2014, the CN algorithm is developed for histopathological images. The algorithm is evolved from another algorithm based on nonlinear mapping of pixel classification.

3) Macenko: This algorithm was developed by Macenko et al. [20] in 2009 with the aim of stain normalising histopathological images, based on the assumption of two dyes used in the staining of the image.

4) Reinhard: This method was proposed by Reinhard et al. [21] in 2001 with the purpose of transferring the general colour content of a synthetic image to another image.

D. Artificial intelligence in digital pathology

According to Niazi et al. [4], artificial intelligence (AI) is already being used within the field of medicine, mainly within radiology and cardiology. However, within digital pathology, AI is still an active field of research. In contrast to radiology, histopathological images are larger, contain colour information, and show no anatomical information; the task of studying histopathological slides is therefore more difficult.

With the assistance of artificial intelligence in pathology, pathologist can more easily get an overview when studying slides. However, creating models, such as convolutional neural networks, is challenging. As many pathologist can disagree about the information presented on the same slides, and cohesiveness regarding colour results after staining is lacking, it is difficult to create accurate training material for the models.

III. Background

A. EfficientNet

The EfficientNet architecture was first outlined and proposed in 2019 by Tan et al. [22]. In Figure 2, the efficiency of the EfficientNet-B0 architecture used in this project is outstanding compared to the other CNN architectures. The

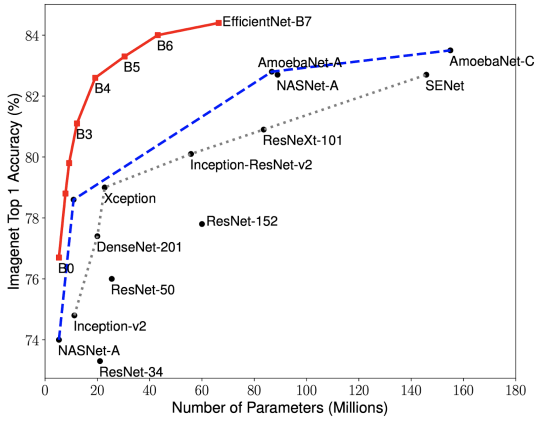


Fig. 2. A comparison of the performance of different architectures. EfficientNet B0 is outperforming the architectures with the same number of parameters. Source: Figure adapted from [22].

architecture consists mainly of *mobile inverted bottlenecks (MBconv)*, as can be seen in Table I. The operation MBConv is an expansion of an ordinary convolutional operation and is explained in-depth by Sandler et al. [23].

TABLE I
THE EFFICIENTNET ARCHITECTURE FOR THE B0

Operator	*Resolution*	*#Channels*	*#Layers*
Conv3×3	224×224	32	1
MBConv1, $k3 \times 3$	112×112	16	1
MBConv6, $k3 \times 3$	112×112	24	2
MBConv6, $k5 \times 5$	56×56	40	2
MBConv6, $k3 \times 3$	28×28	80	3
MBConv6, $k5 \times 5$	14×14	112	3
MBConv6, $k5 \times 5$	14×14	192	4
MBConv6, $k3 \times 3$	7×7	320	1
Conv1 $\times$ 1 & Pooling & FC	7×7	1280	1

The convolution layer is used to extract features from an image. As described by Indolia et al. [24], a convolution layer can consist of multiple kernels, often sized 3×3 or 5×5. These kernels iterate over the image. For each iteration, the scalar product of the kernel is stored in a tensor, becoming the input for the next layer. A visual representative of this operation can be seen in Figure 3.

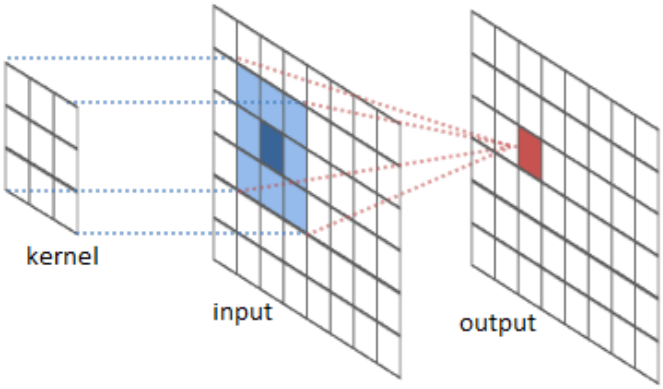


Fig. 3. A representative image of a convolution operation. The kernel scans a section of a layer (blue), and the scalar product of the section becomes a part of the next layer (red). Source: Figure adapted from [25].

B. The Pipeline

The pipeline was created by Tobias Estreen [7] as a part of the ongoing collaboration between KTH and KI. It can be found as open source code on Github [26].

1) Swedish National Infrastructure for Computing - SNIC: The pipeline was run on SNIC's machine-learning oriented, high performance computing cluster Alvis. For the training of the models, Nvidia's V100 Tensor Core graphics cards were utilised, and for smaller computations Nvidia's T4 graphics cards were used. The hardware is described in detail by the SNIC centre *C3SE Chalmers University of Technology* [10].

2) Preparing the datasets for training, validation and testing: Due to EfficientNet being limited to classifying images showing only one cell, the dataset's images were divided into cell-sized images. From every full image, multiple 32×32 px images were generated. Each colour channel was transformed from 0-255 to 0-1. The 32×32 px images were then upscaled using bicubic interpolation to a size of 224×224. Gaussian noise with a standard deviation of 0.1 was added to the training images.

3) Cross-validation: According to Brownlee [27], cross-validation in an effective tool for decreasing possible resulting bias of a model. Cross-validation was used for training the models; it required the input of a number of *splits* and *epochs*. The method divided the input images into a "training set" and a "validation set" uniquely for each split. Each split started with an untrained model that was inserted into the training and validation algorithm an epoch number of times. In this pipeline, a stratified K-Folds cross-validator was used and is described in [28].

C. Ensemble

An ensemble is based on a combination of multiple models. According to Rokach [29], the models must be diverse in order to make the ensemble efficient.

Diversity can be achieved in a variety of ways, e.g. by using different ML algorithms, or by training on different but closely-related datasets. In the two ensembles constructed in this study, the diversity comes from training on closely-related datasets.

For each cell, all models return a weighted likelihood prediction of the cell type. The cell type with the highest value is the predicted cell type of the cell. For example:

$$\begin{bmatrix} -5.4 & 3.5 & 0.3 & -0.1 \end{bmatrix}$$

In this case, the model has predicted the cell to be of the second class (*inflammatory*, *lymphocyte*, *fibroblast and endothelial*, and *epithelial*), which are the *lymphocytes*.

For the ensemble to predict the cell type of a cell, it takes the weighted likelihood prediction of each model and super-positions them into its own weighted likelihood prediction. For all model predictions to be considered equally important, the models' weighted likelihood predictions were transformed to values from 0-1 before the super-positioning, using *softmax*:

$$softmax(x)_i = \frac{exp(x_i)}{\sum_j exp(x_j))}.$$

For example:

$$\begin{bmatrix} -5.4 & 3.5 & 0.3 & -0.1 \end{bmatrix}$$
$$\longrightarrow \begin{bmatrix} 0.00013 & 0.94 & 0.038 & 0.026 \end{bmatrix}.$$

According to Rokach [29], multiple methods for determining the ensemble output were possible, including majority voting and super-positioning the probabilities. Of these two methods, Cheng et al. [30] reported that the latter outperforms the former.

D. Performance Evaluation Metrics

The results will be evaluated through *accuracy, precision, recall and F1-score* metrics as well as *confusion matrices*. The following explanation is based on Mohajon [31]. The metrics are defined using the following terms: *true positive (TP), true negative (TN), false positive (FP)* and *false negative (FN)*. *True* means that the prediction corresponds to the truth determined by the labels, and *false* means that the prediction was incorrect.

For example, what if we have an image of an *epithelial cell* that we want a model to correctly predict?

True Positive (TP) - A TP would mean that the model has correctly predicted the cell as an epithelial cell.

False Negative (FN) - A FN would mean that the model has classified the cell as something other than epithelial, that it is "not an epithelial cell". This prediction is false, as it is an epithelial cell.

True Negative (TN) - A TN would occur if the model predicted this is "not a lymphocyte cell" - or any other kind of cell that is not an epithelial cell - as such a statement would be true.

False Positive (FP) - A FP would occur if the model predicted "this is a lymphocyte" - or again any other kind of cell that is not an epithelial cell - as such a statement would be false.

1) Accuracy: Accuracy answers the question: how many of the total number of cells (all cell types) were correctly predicted?

$$\textbf{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

2) Precision: Precision measures how many of the predictions of a *specific* cell type are true. For example: how many of the cells that were predicted as epithelial cells were actually epithelial cells?

$$\textbf{Precision} = \frac{TP}{TP+FP} =$$

$$= \frac{\text{Number of correctly predicted cells of a specific cell type}}{\text{Number of predicted cells of a specific cell type}}$$

3) Recall: Recall is the *accuracy of a specific cell type*. It measures how many cells of a specific cell type were correctly predicted. For example: how many of the epithelial cells were correctly predicted?

$$\textbf{Recall} = \frac{TP}{TP+FN} =$$

$$= \frac{\text{Number of correctly predicted cells of a specific cell type}}{\text{Number of cells of a specific cell type}}$$

4) F1-score: The F1-score is the mathematical "harmonic mean of precision and recall". The F1-score is often used when the classes (cell types) are *unbalanced*, i.e. different number of objects in each class.

$$\textbf{F1-score} = \frac{2 \cdot \mathit{Precision} \cdot \mathit{Recall}}{\mathit{Precision} + \mathit{Recall}} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$$

If 100 people are tested for cancer and 10 of them have cancer, and we identify 5 of the people with cancer and say that the rest are healthy, then we have an accuracy of 95%, precision of 100%, but a recall of 50%. This would give us an F1-score of roughly 65%. An accuracy of 95% sounds really good, but identifying half of the cancer sick people as healthy is disastrous for those people. The F1-score is therefore a more relevant metric than accuracy because of the unbalanced "dataset" and the fact that the *false negatives* come at too high a cost. The aim of all of the metrics is of course 100%, but this is hard to achieve. In this study it is vital to identify the *inflammatory* and *lymphocyte* cells, as they are important factors when making a GVHD diagnosis, as explained by Tollemar et al. [8].

5) Confusion Matrix: The question that we can't answer using accuracy, precision, recall and F1-score is: when the model misclassifies the cell type, what does it predict instead? A confusion matrix answers this question. It is a matrix with the truth on one axis and prediction on the other, i.e. the matrix shows how the classes are being classified. An example

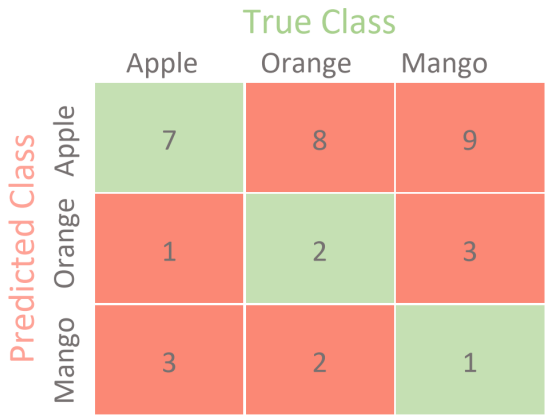


Fig. 4. A confusion matrix of three different types of fruit: apple, orange and mango. The columns represent the actual type of fruit and the rows represent the predicted type of fruit; the green cells are therefore correctly classified fruit and the rest of the cells are misclassified. For example, column 1 row 2 shows that one apple has been misclassified as an orange. Source: Figure adapted from [31].

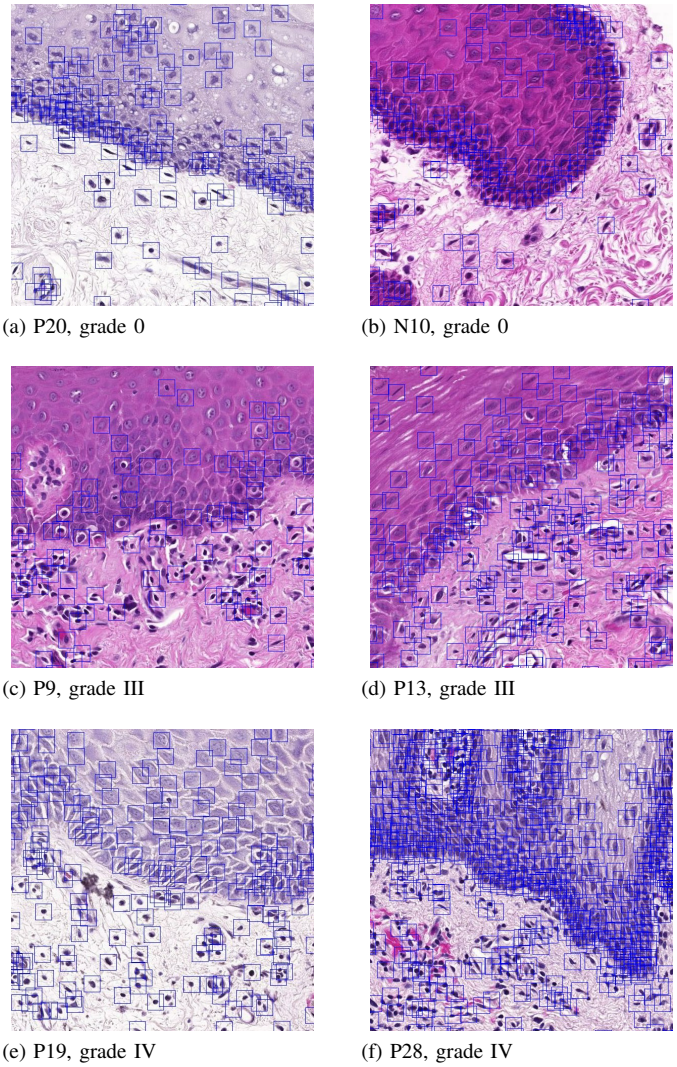
of a confusion matrix can be found in Figure 4, containing three different fruit classes. The Figure has the true classes on the x-axis, which means that the recall of a fruit is the "green value" of the fruit's column divided by the sum of the column. For example, the recall value for the orange is **Recall orange** = $\frac{2}{8+2+2} \approx 0,17$. The prediction is on the y-axis, therefore the precision of a fruit is the green value of the fruit divided by the sum of the fruit's row, for example **Precision orange** = $\frac{2}{1+2+3} \approx 0,33$. That gives the oranges **F1-score** = $\frac{2 \cdot 0.33 \cdot 0.17}{0.33+0.17} \approx 0.22$. The accuracy of the matrix is **Accuracy** = $\frac{7+2+1}{7+8+9+1+2+3+3+2+1} \approx 0,28$. The matrix shows that most of the fruit is being classified as apples.

IV. Material

A. The Dataset

For this study, a subset of H&E stained tissue samples compiled by Tollemar et al. [8] was used. The subset consisted of WSI images of samples from six different patients referred to as *P9, P13, P19, P20, P28* and *N10*. The tissues' histological severity had been graded from 0 (healthy) to IV (severe), based on table II in [8]. The grading was partly related to the number of inflammatory cells and lymphocytes, as well as histologically specific features. The training/validation set consisted of samples from patients P20, P9 and P19, while the test set contained images from N10, P13 and P28. Patients P20 and N10 were healthy (G0), patients P9 and P13 had

moderate GVHD (GIII) and patients P19 and P28 had severe GVHD (GIV). Cropped images from these patients are shown in order of severity in Figure 5. The training/validation set and the test set were thereby balanced in terms of severity.



(a) P20, grade 0 (b) N10, grade 0

(c) P9, grade III (d) P13, grade III

(e) P19, grade IV (f) P28, grade IV

Fig. 5. A visualisation of labels on unnormalised images from the training/validation set vs. the test set. The images on the left side are training/validation images, and the ones on the right are test images. They are placed in order of severity, with the first row being healthy, the second row being moderately inflamed and the third row being severely inflamed.

Images aren't enough to train a model. In order for the computer to understand what it's looking for, pathologists from KI have manually labelled the images with cell location and type. The set of labels labelled from scratch by the pathologists were called *the ground truth*. The ground truth labels do not include all cells in each image, but most of them.

Using active learning, the ground truth set of labels and the dataset were fed into the pipeline [7]. The pipeline added additional labels to the training/validation set. The pathologists then examined the output of labels to correct the additional, computer-generated labels. This new set of labels was referred to as the *reannotated* set. Note that the reannotated set only consisted of the patients used for training/validation, not testing.

At the time of this study, the pathologists hadn't corrected all of the labels for P20, so only the finished label files were included in the reannotated set. There were 21 images of P20 in the ground truth label set but only 10 images of P20 in the reannotated label set.

The sets of labels consisted of different numbers of labels for each cell type. Table II shows the cell-type distribution of the ground truth training/validation set (P20, P9, P19), while Table III shows the cell-type distribution of the reannotated set (P20, P9, P19).

The reannotated set entailed fewer labels in total than the ground truth training/validation set, as not all of the images for P20 were included. The difference in P20 images included could be why the reannotated set had a more balanced distribution than the ground truth training/validation set, as P20 is considered healthy and therefore consisted of very few inflammatory cells and lymphocytes.

TABLE II
DISTRIBUTION OF *the ground truth* LABELS FOR EACH PATIENT IN THE TRAINING/VALIDATION SET IN ORDER OF SEVERITY. THE CELL-TYPE ABBREVIATIONS ARE: INFLAMMATORY (INF), LYMPHOCYTE (LYM), FIBROBLAST AND ENDOTHELIAL (FIB & END), AND EPITHELIAL (EPI).

	Cell type				
Patient	Inf [%]	Lym [%]	Fib & End [%]	Epi [%]	Total #
P20	3.20	7.13	28.5	61.1	13121
P9	11.7	32.0	22.5	33.7	1261
P19	14.8	26.3	33.9	25.1	3009
Total Set	5.82	12.2	29.0	52.9	17391

TABLE III
DISTRIBUTION OF *the reannotated* LABELS FOR EACH PATIENT IN THE TRAINING/VALIDATION SET IN ORDER OF SEVERITY. THE CELL-TYPE ABBREVIATIONS ARE: INFLAMMATORY (INF), LYMPHOCYTE (LYM), FIBROBLAST AND ENDOTHELIAL (FIB & END), AND EPITHELIAL (EPI).

	Cell type				
Patient	Inf [%]	Lym [%]	Fib & End [%]	Epi [%]	Total #
P20	3.11	4.37	30.8	61.7	9190
P9	8.31	24.0	40.5	27.2	2250
P19	9.94	23.4	41.0	25.7	3794
Total Set	7.12	20.9	37.4	38.2	15234

TABLE IV
DISTRIBUTION OF *the ground truth* LABELS FOR EACH PATIENT IN THE TEST SET IN ORDER OF SEVERITY. THE CELL-TYPE ABBREVIATIONS ARE: INFLAMMATORY (INF), LYMPHOCYTE (LYM), FIBROBLAST AND ENDOTHELIAL (FIB & END), AND EPITHELIAL (EPI).

	Cell type				
Patient	Inf [%]	Lym [%]	Fib & End [%]	Epi [%]	Total #
N10	34.0	2.26	30.9	63.4	2794
P13	16.6	13.9	39.3	30.2	2004
P28	10.1	27.5	24.6	37.8	7220
Total Set	9.63	19.3	28.5	42.5	12018

The ground truth test set shown in Table IV has the highest ratio of inflammatory cells and the lowest ratio of fibroblast and endothelial cells when compared to the training/validation sets, as well as the lowest total number of labels.

The number of images in the test set and their ground truth labels are shown in Table V. The number of labels in each

TABLE V
THE RANGE AND NUMBER OF GROUND TRUTH LABELS FOR THE IMAGES OF EACH PATIENT IN THE TEST SET.

		Number of labels		
Patient	#images	Min	Max	Mean
N10	7	26	971	399
P13	4	679	829	501
P28	4	290	3612	1805

image varies greatly. The number of labels in the images increased with each patient (N10, P13, P28). The Table also shows that ca. half of the images are of healthy tissue samples (N10).

B. Ethical approval

Approval for using the dataset used in the ongoing collaborative project between KTH and KI, in which our dataset was included, was granted by the Swedish Ethical Review Authority (Etikprövningsmyndigheten), Dnr: 2019-01259. Our supervisor R. Sugars was the main applicant and responsible researcher. The process of anonymising the dataset was executed at KI before it was sent to KTH, all patient information was strictly excluded.

V. METHOD

A. Methodology

The structure of the study is based on replicating Pontalba et al. [6]. The same CN toolbox has been used, as well as some of the CN algorithms. The methods differ slightly as this study is a multi-classification problem and [6] is a segmentation problem. The ensemble method in this study is therefore similar to [6] but adapted for the multi-classification problem.

The dataset is biased, as shown in Tables II, III and IV. With this is mind, the F1-score should be considered to be of greater importance than the accuracy.

B. Colour normalisation

To normalise the dataset, a *Stain Normalisation Toolbox* [32] created by *The University of Warwick* was used. The toolbox uses a Matlab implementation of Histogram [19], Khan [17], Macenko [20] and Reinhard [21]. Image P9-3-1, shown in Figure 6, was selected as the target image by studying the quality of the general colour content and the contrasts in the image. The choice was confirmed by an experienced pathologist. A well-defined folder system was created to avoid confusion about which images were filtered by which algorithms. Once a target image had been selected and a folder system created, all of the source images (the dataset) were run through the CN algorithms.

C. Preparing the pipeline

1) Adaptation of the pipeline: The pipeline had hard-coded which images were to be used, what was saved and where. To be able to execute the study, these aspects were modified. The adapted pipeline now accepted images as input and saved the performance metrics and checkpoints necessary. Some of the code in the pipeline was reused when recreating the models.

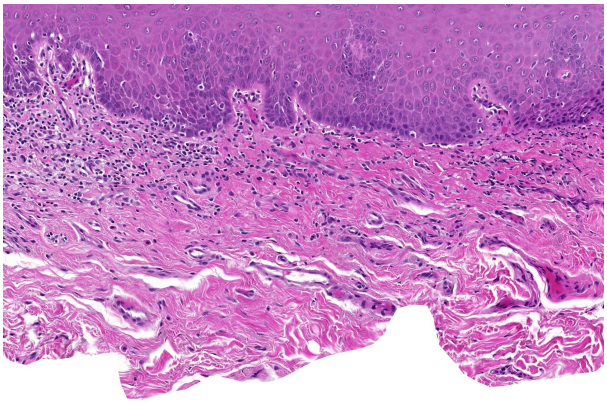


Fig. 6. The target image (P9-3-1) used by the CN algorithms. It is very pigmented with clear contrasts.

2) Running a job: A job is a script sent to Alvis, telling it what to run in the pipeline. The job assigns what dataset should be used, which labels, the hyperparameters, how long the job should run for, where to save different outputs etc. For example, the dataset made up from images colour normalised with the CN algorithm *Khan* could be sent together with the ground truth set of labels, for five splits, with a hundred epochs; among other input-values. A *checkpoint* is required to recreate the model from an epoch, these were saved and easily retrieved at a later date.

D. The Ground Truth Experiment - G0

A job was sent for each of the datasets created from the CN algorithms *Histogram, Khan, Macenko* and *Reinhard*, as well as the unnormalised dataset. The ground truth labels were used as well as the hyperparameters in Table VI.

TABLE VI
HYPERPARAMETERS FOR G0 AND G1.

split	initial learning rate	weight decay	batch size	pretrained
5	0.01	$5e-5$	32	yes
epoch	image size	momentum	advprop	
100	224	0.9	-val -e	

E. The Reannotated Experiment - G1

This experiment followed the same procedure as G0. It used the same datasets and the same hyperparameters, shown in Table VI. The reannotated set of labels were used for training/validation, and patients N10, P13 and P28 from the ground truth set of labels were used for the testing, as with G0. The point of including G1 in this study was to determine how the labels created from active learning affected the results.

F. Ensemble

Two ensembles were created in this study. The first one used all of the best models based on each dataset, a total of five models including a model trained on the unnormalised dataset. This ensemble was referred to as *Ensemble I*. The second ensemble consisted of the three models with the highest

test accuracy, and was referred to as *Ensemble II*. Pontalba et al. [6] used all of its models in its ensemble, which makes it comparable to Ensemble I. It used one additional CN algorithm that was not included in this study.

G. Testing

When a job had finished, multiple checkpoints had been saved. The *mean* and *standard deviation* of the checkpoints with the highest accuracy from each split were calculated. For example, if there were five splits, then one checkpoint from each split, five in total, were used in the calculation.

The best checkpoint of each job, chosen based on its test accuracy, was recreated. The recreated models were tested again on the test set. For each image, a confusion matrix was retrieved and the *precision*, *recall* and *F1-score* of each cell type were calculated from the matrix. For each model, the fifteen confusion matrices (one for each image) were compiled into one matrix from which the accuracy was calculated and a heatmap produced. The confusion matrices had the truth on the y-axis and the predictions on the x-axis. Note that these axes were the opposite from the axes in the fruit example in Figure 4.

VI. Results

A. Colour normalisation

Samples of the colour-normalised images are shown in Figure 7. The images in the figure are cropped, but the full images were used in the experiments. The images were in varying shades of pink, purple and blue. The reference image was the image used as the target image by the CN algorithms.

A general sense of the different CN methods is given by Figure 7. Histogram is transferring the colour of the target image onto the dataset well, but some of the contrasts in the images were lost. Khan was very consistent across the images, giving an overall pink colouration and contrasts that retained the distinctiveness of the features. The same was true for Macenko, albeit the paler-stained images (P20-7-1 and P19-1-1) had a slightly blue hue that was not present in the target image. Finally, Reinhard performed similarly to Khan, although it accentuated the strength of the original staining more than Khan.

1) Image size: The images increase in size from being normalised by one order of magnitude (roughly 1 MB to 10 MB). The unnormalised dataset and the new datasets were all supposed to consist of .tif-files, but upon investigation it seemed the unnormalised dataset actually consisted of .jpg files, which are compressed. When the images were saved using WSI, they were so big that it was not possible to store them in their full sizes. They were therefore saved as compressed files. This seems to be the root cause of the different sizes of the images in the datasets. Apart from the increase in size, the only CN algorithm that showed signs of malfunction was Macenko.

2) Macenko: When applying the *Macenko* CN algorithm to the dataset, the results of some images were unexpected. There were artefacts in the form of discolourations in yellow and deep blue, seemingly at random, as shown in Figure 8. Despite the discolourations of the dataset, Macenko performed surprising well overall. As is not possible to determine how much the artefacts are affecting its performance, the results may not be comparable to Macenko results in other papers that did not present any artefacts.

The discolouration was partially investigated by applying Macenko to two already normalised images (Khan and Reinhard), which were therefore ten times bigger than the unnormalised images. Those images did not present any artefacts, which implies that the artefacts stem from a lack of information; but because of the limited data, no conclusion of substance could be drawn from this.

B. EfficientNet

Moving forward, models and results will be referred to by the names of the CN algorithms used to create their datasets. For instance, a model trained on images normalised by Khan will be referred to as Khan. We therefore have the following models: Unnormalised, Histogram, Khan, Macenko, Reinhard, Ensemble I (based on all of the models) and Ensemble II (based on Unnormalised, Khan and Reinhard).

For every job in G0, the epoch with the highest test accuracy in each split was retrieved and the mean and standard deviation (st. d.) calculated. The results are shown in Table VII.

All of the test accuracy means lay within eight percentage points of each other. Unnormalised gave an average performance, above Macenko and Histogram but below Khan and Reinhard. The models named after CN algorithms had a higher standard deviation than Unnormalised. This is unexpected as the CN algorithms aimed to decrease the colour variations of the images. All of the models had a high validation mean, above 90%, and a lower testing mean around 60%, suggesting that they're overfitted.

TABLE VII
G0: The model with the highest test accuracy was selected from each split. The mean and standard deviations of the models were calculated.

	Test accuracy [%]		Validation accuracy [%]	
Method	Mean	St. d.	Mean	St. d.
Unnormalised	57.30	0.35	94.25	3.59
Histogram	55.13	0.88	93.78	5.78
Khan	60.58	1.50	91.29	7.79
Macenko	54.34	0.53	95.04	2.66
Reinhard	61.26	1.13	97.95	0.52

The same results were calculated for G1 and are shown in Table VIII. Unnormalised and Khan have significantly increased their respective test accuracy means, compared to Table VII. The models named after the CN algorithms still have a larger standard deviation when compared to unnormalised, but the deviations are smaller than that in Table VII. The results suggest that the reannotated labels increased the test accuracy as well as decreased the deviations stemming from the differences in the datasets.

From G0, the models with the highest test accuracy *per job* were retrieved. The models were rerun over the test sets, creating a confusion matrix *per full image* (e.g. P9-3-1), fifteen in total as there were fifteen images in the test set. This

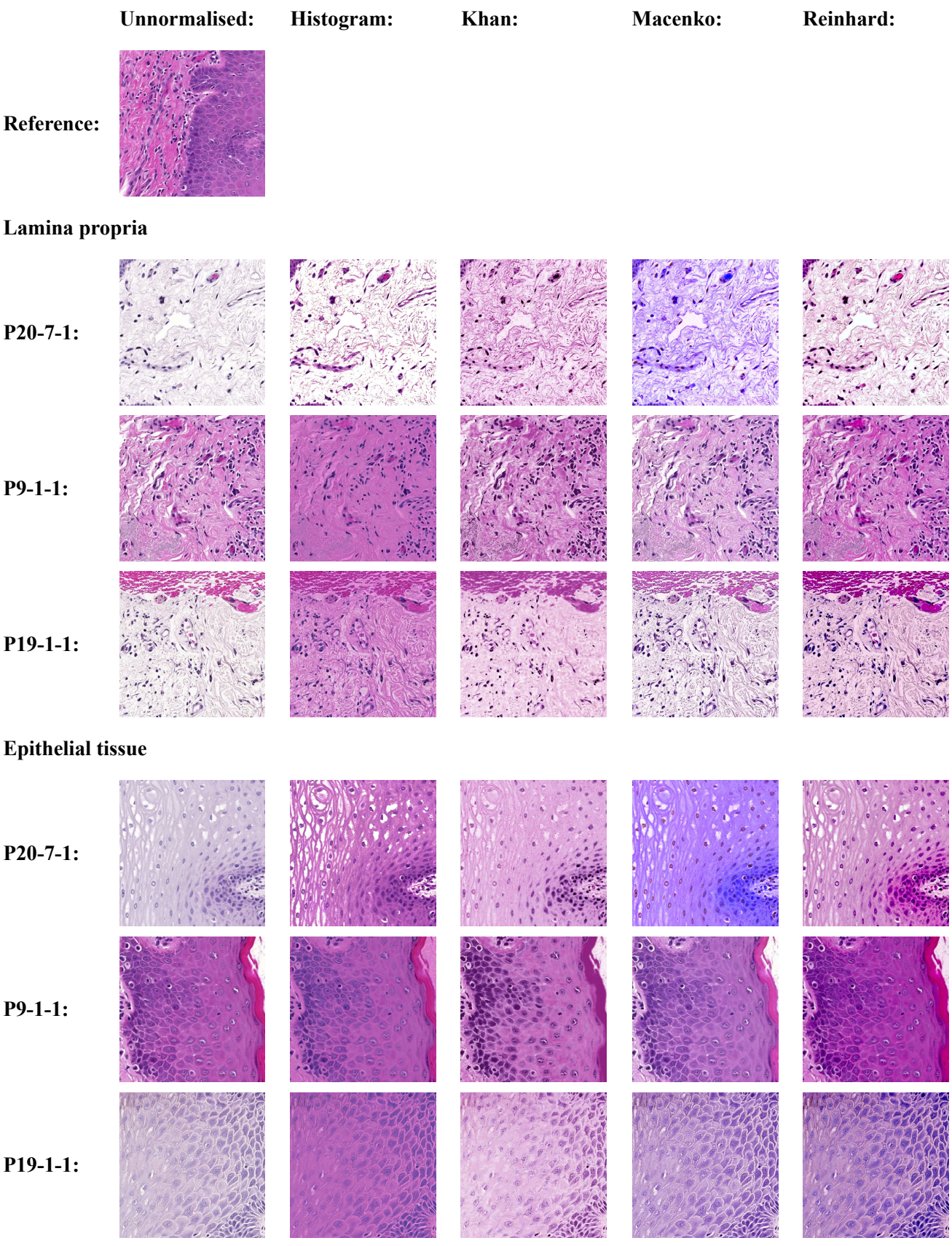
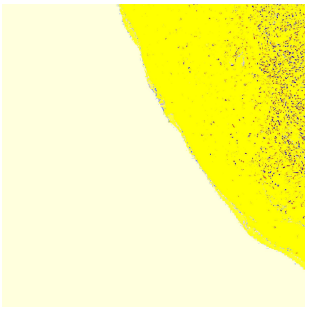
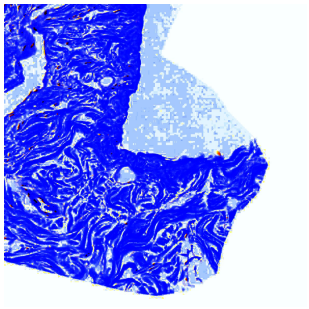


Fig. 7. The effects of the CN algorithms on the training/validation set. Three images were used, one from each patient. The images were colour normalised by the four CN algorithms, and then cropped into images containing epithelial tissue or lamina propria.



(a) A colour normalised image with yellow discolouration (P19).



(b) A colour normalised image with blue discolouration (P20).

Fig. 8. Two samples of images normalised using the Macenko algorithm. The left image has yellow colour artefacts and the right image has blue colour artefacts.

TABLE VIII
G1: THE MODEL WITH THE HIGHEST TEST ACCURACY WAS SELECTED FROM EACH SPLIT. THE MEAN AND STANDARD DEVIATIONS OF THE MODELS WERE CALCULATED.

Method	Test accuracy [%] Mean	St. d.	Validation accuracy [%] Mean	St. d.
Unnormalised	60.59	0.25	97.15	0.47
Histogram	55.07	0.46	95.54	3.51
Khan	65.32	0.73	94.43	3.6
Macenko	55.9	0.71	94.82	4.99
Reinhard	61.42	0.33	97.64	0.56

was repeated for the ensembles. All of the metrics (*accuracy*, *precision*, *recall* and *F1-score*) were calculated from the matrices. The weighted accuracy mean and st. d. of the test sets are shown in Table IX. The weights are the amount of labels used in each image. Macenko and Histogram performed the worst, followed by Unnormalised. Khan and Reinhard performed moderately well, and the ensembles performed the best. The ensembles had a weighted mean of nine to eleven percentage points above Unnormalised. They have a larger st. d. than Unnormalised but a smaller st. d. than most of the other models.

The calculated metrics were made into boxplots and are shown in Figures 9, 10, 11 and 12. The boxes in the boxplots are made up of fifteen values, one for each image in the test set.

TABLE IX
THE WEIGHTED ACCURACY OF THE CHOSEN G0 MODELS ON THE TEST SET.

Method	Weighted Mean [%]	Weighted St. d. [%]
Unnormalised	57	8.4
Histogram	56	9.2
Khan	62	12
Macenko	55	13
Reinhard	63	13
Ensemble I	66	10
Ensemble II	68	11

1) Accuracy: The accuracy of each model on the images in the test set is shown in Figure 9. Ensembles I and II performed the best with high medians and performed relatively consistently. Unnormalised, Histogram and Macenko performed the worst. Unnormalised had the lowest amount of variation.

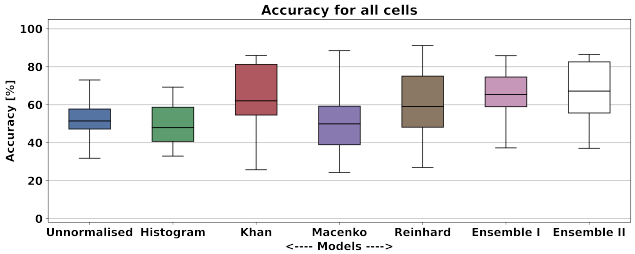


Fig. 9. The accuracy of the models over fifteen test images. The ensembles performed the best, followed by Khan and Reinhard.

2) Precision: The boxplots in Figure 10 show that the models' medians generally increased for each cell type (*inflammatory*, *lymphocyte*, *fibroblast and endothelial*, and *epithelial*). For the inflammatory cells and lymphocytes, the medians were ca. 20 %. The fibroblasts and endothelial cells had precision medians of ca. 60% (with the exception of Khan), and the precision medians of the epithelial cells were around 75%.

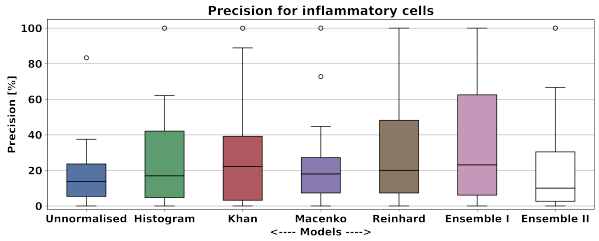
The precision of the inflammatory cells for each model is shown in 10a. Khan and Ensemble I had the highest medians, ca. 22%, while all other models had medians equal to or below 20%. Ensemble II had the lowest median at 10%. The *interquartile range* (IQR) - the coloured box - was the smallest for Unnormalised. Reinhard had the largest IQR, slightly larger than those of Histogram and Khan. The data implied that the IQRs increased due to the colour normalisation.

For the precision of the lymphocytes, shown in Figure 10b, the median was the highest for the four models Reinhard, Ensemble I, Ensemble II and Khan at 22%, 19%, 22%, 23% respectively. Similarly, those models had the largest IQRs, and in that order. Macenko had the lowest median of 10%, closely followed by Unnormalised at 11%. As Reinhard had the highest median, largest IQR and was skewed upwards, as seen in Figure 10b, the model could be said to have the highest precision of lymphocytes.

Regarding the precision of the fibroblast and endothelial cells, Khan had the highest median at 73%, followed by Ensemble I and Ensemble II at 53% and 54% respectively. Even though Khan was skewed downwards, as seen in Figure 10c, and had the largest IQR, the data suggests that it generally had the highest precision of fibroblast and endothelial cells. It is also worth noting that Reinhard and Unnormalised performed very similarly, both in terms of median (ca. 45%) and IQR.

The last cell type, and the most frequently labelled in the test set, was the epithelial cell type. Khan had yet again the highest median at 84%, closely followed by Ensemble II at 82%. Ensemble II had a lower IQR compared to Khan, which implied that Ensemble II was more reliable. This is clearly illustrated in 10d. The precision of Unnormalised and Macenko were quite comparable, with medians at 71% and 70% respectively, and with similar IQR.

Khan, Reinhard and Ensemble I consistently achieved great precision for the cell types, and Unnormalised always performed worse than or equal to them. Ensemble II had the poorest precision of the inflammatory cells on a par with Unnormalised, showed average precision for the lymphocytes and fibroblast and endothelial cell types, and performed better



(a) Precision for inflammatory cells. All models have medians at or below 22%. Khan and Ensemble I had the highest median, while unnormalised had the lowest. At the same time, the IQR is lowest for Unnormalised and Macenko.

(b) Precision for lymphocyte cells. The medians were comparable to those of the models' precision for inflammatory cells. Once again, Unnormalised and Macenko are similar, but the medians which were largest i.e. Khan and the ensembles are similar to that of Reinhard.

(c) Precision for fibroblast and endothelial cells. Khan performed the best, while Reinhard and Unnormalised performed the worst.

(d) Precision for epithelial cells. Ensemble II had the lowest IQR and the next highest median closely behind Khan.

Fig. 10. The precision scores from each test image compiled into one boxplot for each cell type.

than almost all other models for the precision of epithelial cells.

3) Recall: Beginning with the recall of the inflammatory cells shown in Figure 11a; Histogram, Khan, Reinhard and the ensembles all presented low medians of 8.9%, 3.5%, 3.5%, 2.7% and 2.3% respectively. This is equal to or below the median of Unnormalised (16%), which, in turn, is outperformed by Macenko with a median of 20%. Khan, however, is skewed upwards.

The recall of the lymphocyte as shown in Figure 11b is interesting. All of the models outperformed Unnormalised by at least ten percentage points. Furthermore, the CN models had much larger variability than Unnormalised and Ensemble II, implying that ensembling decreased the variability of the models. In addition, Khan performed the best, albeit with large variability.

For the fibroblast and endothelial cells, as shown in Figure 11c, the variability of all the models was similar, with the exception of Khan and Ensemble I which were lower. Ensembles I and II outperformed the other models with their high medians and low variations. Ensemble I had a lower IQR and a higher "lowest quartile" than Ensemble II, making it more reliable. They both had a significant increase in recall compared to their performances on the previous cell types.

All models performed their best recall on the epithelial cells, as shown in Figure 11d. Khan performed the best, closely followed by Ensemble I, Ensemble II and Reinhard, having the highest medians and lowest variability. The three others have lower medians and performed relatively equally in terms of variability.

4) F1-score: The F1-score of each cell type is shown in Figure 12.

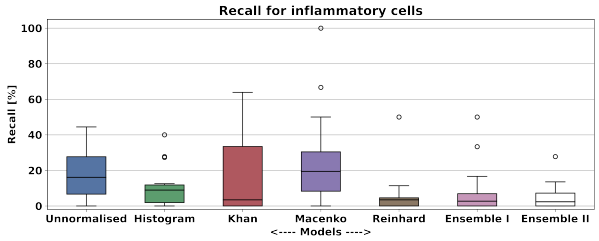
All of the models increased their respective F1-scores for each cell type (*inflammatory*, *lymphocyte*, *fibroblast and endothelial*, and *epithelial*). The models performed relatively equally for the first three cell types, with medians at ca. 7.8%, 22%, 56% and 72% respectively.

When comparing the models' F1-scores for the inflammatory cells, Macenko performed the best with a median of 15%, closely followed by Unnormalised with a median of 11%, both models had low and similar IQRs. Histogram and Khan gave average performances. Reinhard, Ensemble I and Ensemble II performed very poorly, with medians at ca. 5%. The models generally had low medians and low IQRs, with Khan having the largest IQR.

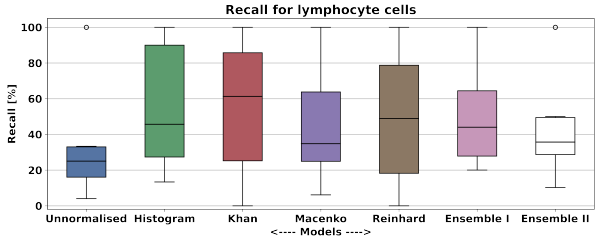
When looking at the F1-scores for the lymphocytes, all of the models had increased both their medians and their IQRs. Ensemble I could be considered the best, based on the facts that its median is the highest (29%), and that the bottom of its box is higher than that of any other, rendering it the most reliable model. Unnormalised was considered the worst and was closely followed by Macenko, with medians at 15% and 12% respectively. Even though Macenko had a lower median than Unnormalised, its IQR showed that it had outperformed Unnormalised in several images.

For the fibroblast and endothelial cells, the ensembles performed the highest. Ensemble II could be considered superior to Ensemble I based on its higher median (64% vs. 62%) and that its bottom quartile is way above that of Ensemble I. The rest of the models had similar medians at ca. 53% and large IQRs, with Histogram and Macenko having the largest IQRs.

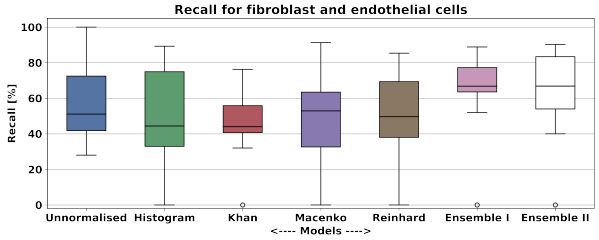
The models' F1-scores for the epithelial cells vary. Ensemble II and Khan perform the best, with medians of 83% and 82% respectively. It could be argued that Ensemble II was more reliable than Khan based on its lower IQR and that its bottom quartile was a lot higher. Unnormalised performs in the middle, i.e. better than Histogram and Macenko but worse than the other models. Histogram and Macenko were significantly



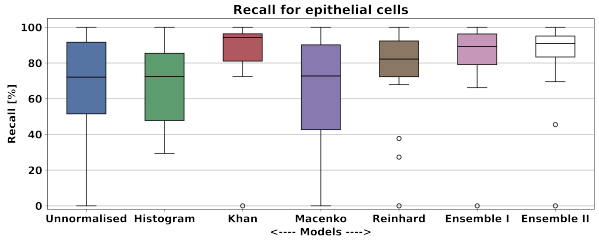
(a) Recall for inflammatory cells. Macenko had the highest median and had generally the highest recall for inflammatory cells compared to all other models. Histogram, Reinhard, Ensemble I and II performed the worst.



(b) Recall for lymphocyte cells. Here Khan was clearly ahead in terms of the median, although it presented a fairly high IQR. Ensemble I outperformed II here, and Unnormalised had the lowest recall.

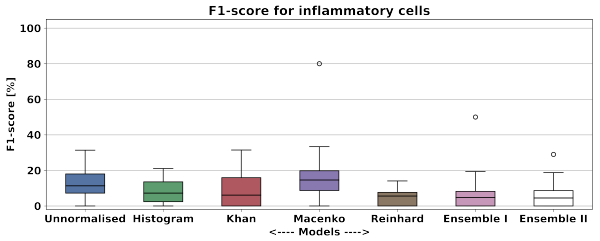


(c) Recall for fibroblast and endothelial cells. Ensemble I and II both had the highest median while the former had the lowest IQR, making it the best model here. Both Histogram and Khan had the lowest median, while Khan had an IQR similar to that of Ensemble I.

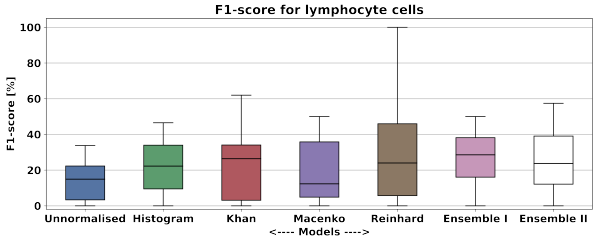


(d) Recall for epithelial cells. Khan had the highest median, and closely followed by the two ensembles and Reinhard.

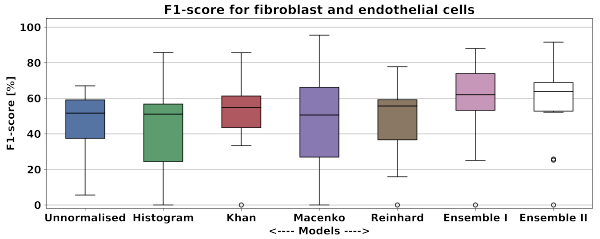
Fig. 11. The recall from each test image compiled into one boxplot for each cell type.



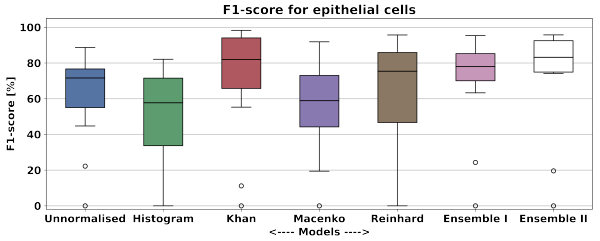
(a) F1-score for inflammatory cells. Macenko were clearly ahead of all the other models, although the median was under 20%. Reinhard and the two ensembles had the lowest F1-score.



(b) F1-score for lymphocyte cells. Here, Reinhard had a large IQR, with its range going all the way to 100%, which no other model´s range does. However, Ensemble I had the highest median and its IQR reached higher than Reinhard making it the best here.



(c) F1-score for fibroblast and endothelial cells. The two ensembles were in the lead and very comparable, with both having low IQRs and high medians. The models with the lowest F1-score were Unnormalised, Histogram and Macenko.



(d) F1-score for of epithelial cells. Here, Ensemble II was clearly the best, with Khan having a similar median with higher IQR. Macenko and Histogram were the two models with the lowest F1-scores here.

Fig. 12. The F1-scores from each test image compiled into one boxplot for each cell type.

worse than all of the other models, with medians of ca. 58% and large IQRs.

When comparing the two ensembles, they tied in regard to the inflammatory cells. Ensemble I performed better than Ensemble II on the lymphocytes, but Ensemble II performed the best on the remaining two cell types.

5) Confusion matrices: The confusion matrices for each model and ensemble are shown in Figure 13.

In general terms, all models classified the fibroblasts and endothelial and epithelial cells best, while the inflammatory cells were mostly misclassified. Reinhard was the best at classifying lymphocytes. This can be seen in Figure 13 by studying the elements on the diagonal. Note that the number of cells for each cell type varied and can be calculated from Table IV. The colours in the heatmaps show the number of cells in that category.

Information regarding misclassification can be determined by examining the rows and columns of the confusion matrices.

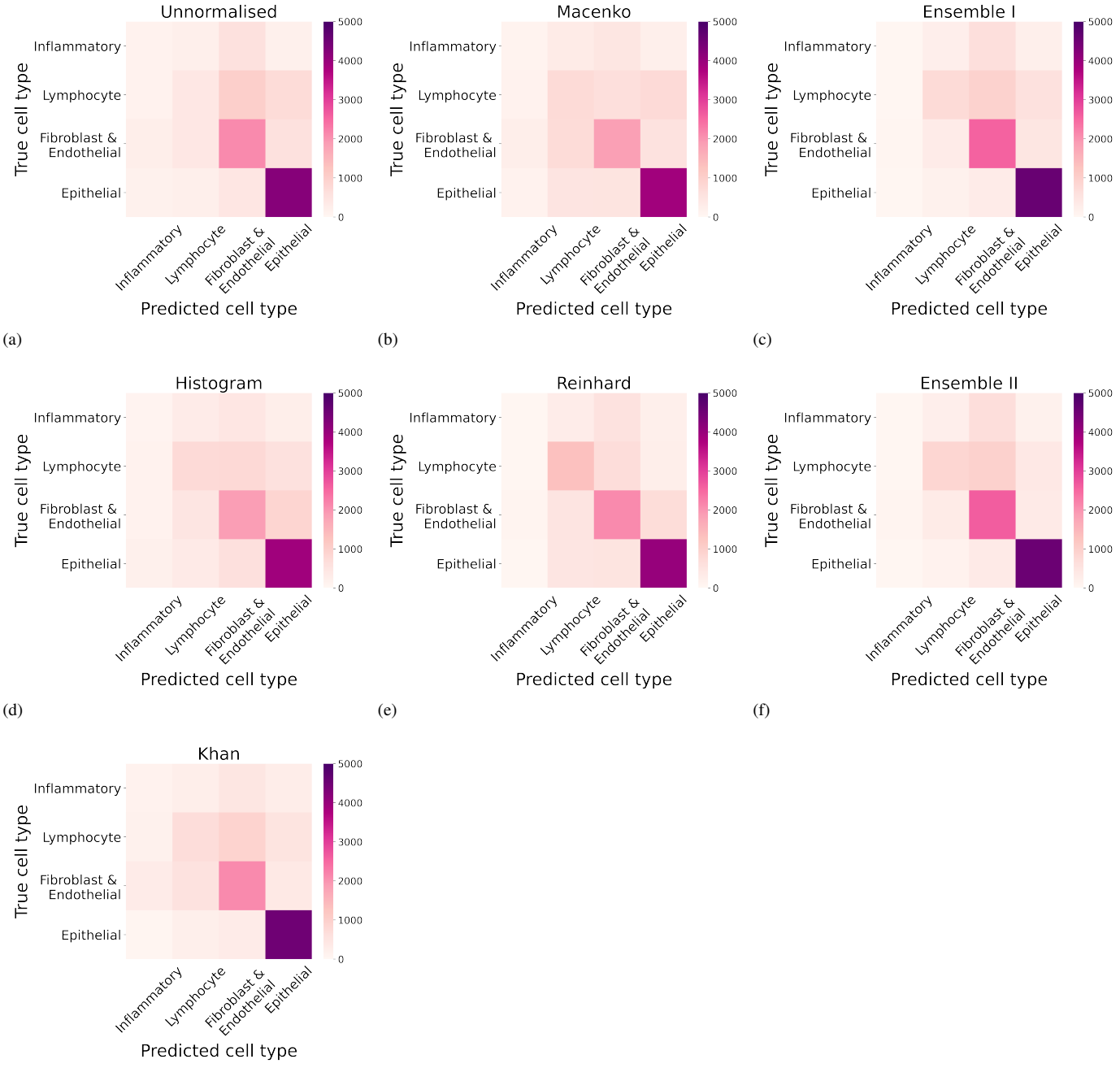


Fig. 13. Heatmaps representing the confusion matrices for each model and the ensembles. The heatmaps show how well the models classified the cell. Looking at the diagonal, all models classified fibroblast and endothelial and epithelial cells fairly well, compared to the classification of the inflammatory cells, which was very poor.

Unnormalised tended to confuse lymphocytes with fibroblast and endothelial and epithelial cells, whilst Histogram, Ensembles I and II and Khan correctly classified lymphocytes almost as frequently as they misclassified them as fibroblasts and endothelial cells. Reinhard was the only model to correctly classify most of the lymphocytes.

All models regularly misclassified inflammatory cells as fibroblast and endothelial cells, with the ensembles doing this the most.

VII. Discussion

The aim of this study was to evaluate how colour normalising histopathological images and ensembling affected the performance of a convolutional neural network. The results indicate that the combination of CN and ensembling *improves* the CNN's performance. As the pathologists' need for a more effective and partly automated workflow grows, at the same time as the problem with variations in staining remains, the use of CN and ensembling could alleviate these issues and assist pathologists in their work.

A. Comparing the test results to Pontalba et al. [6]

One of the conclusions drawn in Pontalba et al. [6] was that the ensemble performed on average the best. This was also the case for the results presented in this report.

As two different kinds of problems were studied, segmentation vs. classification, this alludes that ensembling generalises over different problems in different areas of pathology. To strengthened this hypotheses, further studies would have to be made, using different datasets, CN algorithms, ML models etc. This would also give greater insight into the differences of classification of different cell types.

B. G0 vs. G1

The two experiments differed in three significant ways: the number of labels, the distribution of the labels in regard to cell type (achieved through active learning), and their results.

A comparison of the results in Table VII and VIII showed that G0 had ca. 14% more labels in its training/validation set than G1, but G1 had a more even distribution. It also showed that all models in G1 had either insignificantly changed or improved their test accuracy, and that most models had shown a decrease in their standard deviation. This suggests that the change in the distribution affected the results more than the decrease in number of labels, and was therefore the reason that the test accuracy and standard deviation improved.

Furthermore, the label distribution of the training/validation set shown in Table II was echoed in the metric results shown in Figures 10, 11 and 12, as they indicated that all models performed better cell type for cell type.

The data suggests that, in future studies, the focus should be placed on active learning and evening out the cell-type distribution rather than simply increasing the amount of labels in the dataset.

C. Should other models be used instead of Unnormalised?

For a ML model to be of use to pathologists diagnosing GVHD, it must be able to correctly classify inflammatory cells and lymphocytes. For this reason, the metrics *recall* and *F1-score* are of utmost importance.

Unnormalised and Macenko had recall and F1-scores marginally higher than the other models regarding the inflammatory cells. The fact that the other models performed so much worse implies that the dataset lost important visual features connected to the inflammatory cells when being normalised by the other CN algorithms. That said, the opposite could also be said about Macenko as it performed the best. It is possible that the other models performed so much worse due to the lack of inflammatory cells in the datasets. It could be argued, based on the performances of the models on the other cell types, that the other models would perform better than Macenko and Unnormalised if the number of inflammatory cells had been higher.

For all other cell types, both Ensembles I and II outperformed Unnormalised for all metrics (*accuracy*, *precision*, *recall* and *F1-score*). This suggests that with an increase of inflammatory cells in the datasets, the ensembles could be made to outperform Unnormalised for all cell types.

D. Boxplots vs. heatmaps

Boxplots and heatmaps fundamentally show different information. In the boxplots, all images were considered equally important. The heatmaps show the classification of each cell, thereby considering each *cell* (not image) equally important. Table IV shows that N10 consisted of ca. half of the images in the test set, but that N10 consisted of ca. 23% of all of the labels in the test set, rendering N10 over-represented in the boxplots.

An example of this imbalance can be seen by comparing Reinhard's and Khan's classification of the lymphocyte cells. Studying Figure 11b, Khan appears to be outperforming Reinhard, but when looking at the heatmaps of Reinhard and Khan in Figure 13, they show that only Reinhard is successfully classifying the lymphocytes.

An important difference between the two visualisations is that the boxplots show a *reliability* of a model, as it shows an interval instead of a ratio. The heatmaps can, for example, show the recall ratio of a cell type, but it does not show how it varies between images. For a model to be used in the medical field, it needs to perform consistently, as it is of no use if it performs poorly for some patients.

E. Error factors and biases

When loading the checkpoints used for testing into models, some of the checkpoints performed worse than recorded and some could not be loaded at all. The reason for this is undetermined, but it should be investigated to see whether the problems lie with Alvis, which is plausible as Alvis is a new computing cluster.

In the training/validation set and test set, each histological grade (0, III, IV) is only represented by one tissue sample per set, as shown in Figure 5. As such, each grading is connected to one colour of staining. This could inadvertently have introduced a bias in the model. To what extent, however is unknown.

VIII. Conclusions

In conclusion, the use of CN algorithms and ensembling generally boosted the performance of the models. For the least abundant cell type, the inflammatory cells, the model trained on the original dataset outperformed all other models. The results suggested that the imbalance of the dataset was reflected in the performance of the resulting models, and should therefore be taken into consideration in future studies. The study also showed that the models trained on datasets created by CN algorithms had larger variation. The models based on the datasets created by Khan and Reinhard generally performed better than the original model, and that models trained on datasets created by Histogram and Macenko generally performed the worst.

A. Future studies

The first, and perhaps the most obvious area for a future study, would be to use the models to train *EfficientDet*, as EfficientDet can take in images consisting of multiple cells.

This can actually be built onto this project, as EfficientDet uses EfficientNet as its backbone. The reannotated dataset is also of interest when using EfficientDet as it improved the test accuracy for several models.

Secondly, as stated before, studying the effects of different distributions of datasets on CNNs in digital pathology is very interesting, as an imbalance could decrease the average performance.

Finally, as stated previously, studying the effects of image compression on CN algorithms would be very helpful for the continued advancements of digital pathology. If both the level of compression permitted without affecting the results and how to efficiently store digitised pathological images could be investigated, future work routines within the field should be improved.

IX. Acknowledgements

Sincerely thanks to Karl Meinke, Rachael Sugars, Aravind Ashok Nair and Helena Arvidsson.

The computations/data handling were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) at C3SE Chalmers University of Technology partially funded by the Swedish Research Council through grant agreement no. 2020/33-67.

Research funding from ALF Medicine and SOF Clinical Odontological Research Funding for the digital pathology study.

References

[1] A. T. Feldman and D. Wolfe, "Tissue processing and hematoxylin and eosin staining," *Methods in Molecular Biology (Clifton, N.J.)*, vol. 1180, pp. 31–43, Jun. 2014.

[2] M. R. Wick, "The hematoxylin and eosin stain in anatomic pathology-An often-neglected focus of quality assurance in the laboratory," *Seminars in Diagnostic Pathology*, vol. 36, no. 5, pp. 303–311, Jun. 2019.

[3] H. Lyon and R. Horobin, "Standardization and standards for dyes and stains used in biology and medicine," *Biotechnic & Histochemistry*, vol. 82, no. 1, pp. 1–11, Jul. 2009. [Online]. Available: https://doi.org/10.1080/10520290601116590

[4] M. K. K. Niazi, A. V. Parwani, and M. N. Gurcan, "Digital pathology and artificial intelligence," *The Lancet. Oncology*, vol. 20, no. 5, pp. e253–e261, May 2019.

[5] J. Pallua, A. Brunner, B. Zelger, M. Schirmer, and J. Haybaeck, "The future of pathology is digital," *Pathology - Research and Practice*, vol. 216, no. 9, p. 153040, Jun. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0344033819330596

[6] J. T. Pontalba, T. Gwynne-Timothy, E. David, K. Jakate, D. Androutsos, and A. Khademi, "Assessing the impact of color normalization in convolutional neural network-based nuclei segmentation frameworks," *Frontiers in Bioengineering and Biotechnology*, vol. 7, p. 300, Nov. 2019. [Online]. Available: https://www.frontiersin.org/article/10.3389/fbioe.2019.00300

[7] T. Estreen, "Epithelial Layer Boundary Detection Using Graph Convolutional Networks for Digital Pathology," Master's thesis, KTH Royal Institute of Technology, Stockholm, Sweden, Sep. 2020.

[8] V. Tollemar, N. Tudzarovski, G. Warfvinge, N. Yarom, M. Remberger, R. Heymann, K. Garming Legert, and R. V. Sugars, "Histopathological grading of oral mucosal chronic graft-versus-host disease: Large cohort analysis," *Biology of Blood and Marrow Transplantation*, vol. 26, no. 10, pp. 1971–1979, Jul. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1083879120304018

[9] SNIC. (2021, May) Hpc2n, umeå university. [Online]. Available: https://www.snic.se/resources/compute-resources/kebnekaise/

[10] C3SE. (2021, Apr.) Alvis hardware. [Online]. Available: https://www.c3se.chalmers.se/about/Alvis/#hardware

[11] A. S.-Y. Leong and Z. Zhuang, "The changing role of pathology in breast cancer diagnosis and treatment," *Pathobiology*, vol. 78, no. 2, p. 99–114, Jun. 2011. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3128144/

[12] P. L. Farahani N, Parwani A, "Whole slide imaging in pathology: advantages, limitations, and emerging perspectives." *Pathology and Laboratory Medicine International*, vol. 7, pp. 23–33, Jun. 2015.

[13] B. Smith, M. Hermsen, E. Lesser, D. Ravichandar, and W. Kremers, "Developing image analysis pipelines of whole slide images: Pre- and post-processing," *Journal of Clinical and Translational Science*, vol. 5, pp. 1–33, Aug. 2020.

[14] H. A. Alturkistani, F. M. Tashkandi, and Z. M. Mohammedsaleh, "Histological stains: A literature review and case study," *Global journal of health science*, vol. 8, no. 3, pp. 72–79, Jun. 2015, 26493433[pmid]. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/26493433

[15] A. Bentaieb and G. Hamarneh, "Adversarial stain transfer for histopathology image analysis," *IEEE Transactions on Medical Imaging*, vol. 37, no. 3, pp. 792–802, Dec. 2018.

[16] J. Folmsbee, S. Johnson, X. Liu, M. Brandwein-Weber, and S. Doyle, "Fragile neural networks: the importance of image standardization for deep learning in digital pathology," in *Medical Imaging 2019: Digital Pathology*, J. E. Tomaszewski and A. D. Ward, Eds., vol. 10956, International Society for Optics and Photonics. SPIE, 2019, pp. 222 – 228. [Online]. Available: https://doi.org/10.1117/12.2512992

[17] A. M. Khan, N. Rajpoot, D. Treanor, and D. Magee, "A Nonlinear Mapping Approach to Stain Normalization in Digital Histopathology Images Using Image-Specific Color Deconvolution," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 6, pp. 1729–1738, Jun. 2014.

[18] A. Lain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, New Jersey 07632: Prenitce-Hall, Inc., 1986, pp. 241 – 244.

[19] S. Annadurai and R. Shanmugalakshmi, *Fundamentals of Digital Image Processing*. Delhi 110 092, India: Dorling Kindersley, 2007.

[20] M. Macenko, M. Niethammer, J. Marron, D. Borland, J. Woosley, X. Guan, C. Schmitt, and N. Thomas, "A method for normalizing histology slides for quantitative analysis," *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 1107–1110, Aug. 2009.

[21] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer Graphics and Applications*, vol. 21, pp. 34–41, Oct. 2001. [Online]. Available: https://www.researchgate.net/publication/220518215_Color_Transfer_between_Images

[22] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *CoRR*, vol. abs/1905.11946, 2019. [Online]. Available: http://arxiv.org/abs/1905.11946

[23] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," Jun. 2018, pp. 4510–4520.

[24] S. Indolia, A. K. Goswami, S. Mishra, and P. Asopa, "Conceptual understanding of convolutional neural network- a deep learning approach," *Procedia Computer Science*, vol. 132, pp. 679–688, Jun. 2018, international Conference on Computational Intelligence and Data Science. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918308019

[25] C. Olah. (2014, Jul.) Understanding convolutions. [Online]. Available: https://colah.github.io/posts/2014-07-Understanding-Convolutions/

[26] T. Estreen. (2020, Sep.) Thesisproject. [Online]. Available: https://github.com/testreen/ThesisProject

[27] J. Brownlee. (2020, Aug.) A gentle introduction to k-fold cross-validation. [Online]. Available: https://machinelearningmastery.com/k-fold-cross-validation/

[28] (2020) 3.1. cross-validation: evaluating estimator performance. [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html

[29] L. Rokach, "Ensemble methods for classifiers," *The Data Mining and Knowledge Discovery Handbook*, pp. 957–980, Jan. 2005. [Online]. Available: https://www.researchgate.net/publication/226564652_Ensemble_Methods_for_Classifiers

[30] C. Ju, A. Bibaut, and M. Laan, Apr. 2017.

[31] J. Mohajon. (2020, Sep.) Confusion matrix for your multi-class machine learning model. [Online]. Available: https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826

[32] U. o. W. TIA Centre, Dept of Computer Science. (2015, May) Stain normalisation toolbox. [Online]. Available: https://warwick.ac.uk/fac/cross_fac/tia/software/sntoolbox/

Assessing the Impact of Stain Normalization on a Cell Classification Model in Digital Histopathology

Albert Aillet and Filip Frisk

***Abstract*—In the field of digital histopathology, computer-aided diagnosis of digitized tissue samples with computational algorithms is a rising research field. The tissue samples in this study are stained using chemicals that enhance the recognizability of different tissue structures. This staining can be highly variable, which has an impact on the performance of the computational algorithms. The aim of this project is to assess the use of three color normalization algorithms as a pre-processing step on the KI dataset from a collaborative research project between Karolinska Institutet and KTH Royal Institute of Technology. The color normalization algorithms aim to reduce the color variability of the data. The basis of the study is an implementation of the EfficentNet Convolutional Neural Network classification model, that was adapted for the specific needs of the study. Performance was assessed by firstly applying the color normalization filters to the dataset and training multiple models on each of the filtered datasets. The results from the individually trained models and the combined results with ensemble learning techniques were then analyzed. Our conclusions are clear, stain normalization filters significantly impacts classification performance metrics. The impact depends on the staining qualities of the filters. Ensemble learning techniques present a more robust performance than the individual filters with a performance comparable to the best performing filter.**

***Sammanfattning*—Datorstödd diagnos av digitaliserade vävnadsprov med hjälp av beräkningsalgoritmer inom digital histopatologi är ett aktivt forskningsfält. Vävnadsproven i denna studie har färgats med kemikalier som förbättrar igenkännandet av olika vävnadsstrukturer. Kvaliteten på denna färgningsprocess kan variera, vilket har en inverkan på beräkningsalgoritmernas prestanda. Syftet med detta projekt är att utvärdera användningen av tre färgnormaliseringsalgoritmer som ett förbehandlingssteg på ett dataset från ett samarbetsprojekt mellan Karolinska Institutet och Kungliga Tekniska Högskolan. De använda färgnormaliseringsalgoritmerna syftar till att minska färgvariabiliteten i datan. Grund för studien är en implementering av klassificeringsmodellen EfficentNet, som anpassades utifrån studiens specifika behov. Prestandan bedömdes genom att först använda varje färgnormaliseringsalgoritm på datasetet och träna flera modeller på var och en av de filtrerade dataseten. Därefter analyserades resultaten från de individuella modellerna och de kombinerade resultaten med ”ensemble learning”-tekniker. Våra slutsatser är tydliga, färgnormaliseringen påverkar significant prestandamätvärdena. Dess inverkan beror på filtrens färgningsegenskaper. ”Ensemble learning” teknikerna ger en mer robust prestanda än de enskilt tränade modellerna som lika bra som det bäst presterande filtret.**

***Index Terms*—Digital pathology, Machine learning, Color normalization**

Supervisors: Rachael Sugars & Karl Meinke

TRITA number: TRITA-EECS-EX-2021:198

I. Introduction

Artificial intelligence and machine learning approaches, specifically deep learning models are part of a rising research field within digital healthcare, especially digital histopathology. The workflows of pathologists have in the past been limited to physical samples and analog microscopes. Recent developments in hardware and software have led to a digitization of this workflow. This opens up for the use of deep learning to provide pathologists with reliable support for diagnostic assessment and treatment decisions [1], [2].

Studies have shown that the need for pathology services is high, especially in low to medium income countries. Such countries have more than average disease cases but a low share of global healthcare resources and poor access to quality pathology and laboratory medicine [3]. Even in western countries the access is not evenly distributed across regions and some severely lack competence [4]. This puts heavy load on the available specialists and creates long waiting times in an already pressured healthcare system.

Since 2018, the Oral Biology and Medicine Group at the Department of Dental Medicine at Karolinska Institutet (KI) and the Theoretical Computer Science Division (EECS school) at KTH Royal Institute of Technology school have an ongoing research project on this topic named *Evaluation of Neural Networks for Digital Pathology on High Performance GPUs*. The overarching aim of the project is to provide clinicians with computer-aided diagnostic support.

Prior to this project, multiple Masters and PhD students have been involved, a dataset has been created and different machine learning techniques have been evaluated. The KI dataset consists of cell types from oral mucosa tissue samples hand-labeled by pathologists at the Department of Dental Medicine at KI. At first two deep learning algorithms, Softmax CNN and RCCNet were investigated and the results were not satisfactory in terms of accuracy [5]. A more computationally intensive deep neural network EfficientNet has been used and trained on the Kebnekaise supercomputer [6] in the Masters thesis *Epithelial Layer Boundary Detection using Graph Convolutional Networks for Digital Pathology* [7]. In this thesis, it was proposed that the color variability from the staining process could explain the misclassifications and weak generalizability. This study aimed to investigate color variability, by using methods from the study of Pontalba et al. [8].

Pontalba et al. [8] found that approaches of combining multiple color normalization filters and using ensemble learning techniques might address some of the problems associated

with color variability for a segmentation task. Similarly to the Pontalba et al. this bachelor thesis investigated the use of color normalization filters as a pre-processing step but for a cell classification model instead of a segmentation model. The impact on performance for models trained on the color normalized datasets was analyzed individually and the results from the individual models were combined using ensemble learning techniques.

II. Background

A. Histopathology

1) General: Histopathology is a field of clinical medicine where diagnosis is based on visual examination by pathologists of tissue samples under a microscope. The visual review of a tissue is often subjective, with great variability in the decision depending on the pathologist and the lab. Manual examination of samples is a laborious and time consuming task, especially if the few field specialists that are already in high demand are required [9].

2) Digital Histopathology: The recent development of the digitization of histological samples has enabled a large number of samples to be scanned and archived digitally. A common process is whole slide imaging (WSI) where tissue samples placed on glass slides are digitally scanned [10]. Digital histopathology encompasses all technologies that use these digital slides to allow for improvements and innovations in the workflow of pathologists [11]. Computational algorithms or more specifically AI algorithms can take advantage of the datasets consisting of tissue samples available for analysis to support the pathologists in the diagnosis process [12]. While pathologists have to take the final decision, the AI can highlight structures of interest in the tissue samples. However, these samples need to be annotated by experts to be of use for the AI algorithms which is a long and time consuming task. Consequently the field suffers from a lack of quality annotated data [13]. Construction of an end-to-end WSI deep learning analysis pipeline that can be used in a clinical setting requires many steps, see Fig. 1.

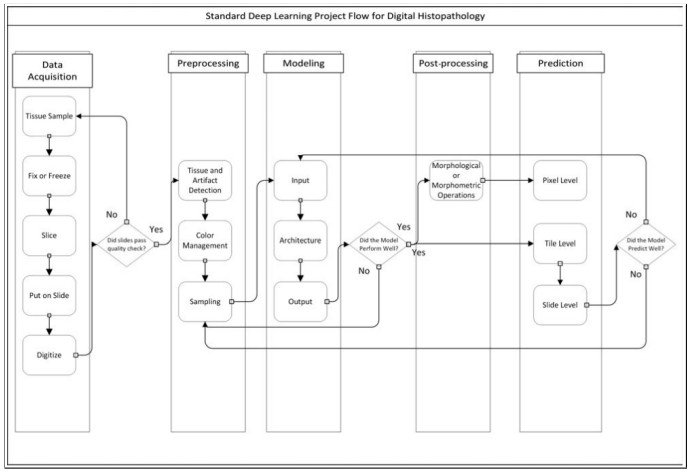


Fig. 1. Typical deep learning project flow in digital histopathology [2]

3) Clinical samples and associated histological grading: The digitized tissue samples in the study *Histopathological Grading of Oral Mucosal Chronic Graft-versus-Host Disease: Large Cohort Analysis* [14] have been histologically graded (G0 to G4). The grading refers to a points-based grading system based on ”intraepithelial lymphocytes and band-like inflammatory infiltrate, atrophic epithelium with basal cell liquefaction degeneration, including apoptosis, as well as separation of epithelium and pseudo-rete ridges” [14]. The grading gives an indication of the histological severity of the tissue sample.

4) Haematoxylin and eosin tissue sample staining and digitization process: Before a tissue sample can be scanned and digitized or observed directly by a pathologist, it has to go through a number of preparation steps to preserve its structure and have an appearance that facilitates the diagnosis of the pathologist [12]. One of the main steps is the staining process. After the initial processing, most tissues and cells are transparent under the microscopy [15] and staining is used to reveal the anatomical features of the tissue structure for visual examination. One of the most common staining processes is Haematoxylin and Eosin (H&E) staining [16]. Eosin is acidic and negatively charged and stains structures like the cytoplasm and extracellular matrix in a red or pink color. Haematoxylin is basic and stains structures like the nuclei in a purple or blue color [17]. After the cut section have been exposed to these two stains they present visually recognizable features that are easier for the pathologist to identify. However, this staining is highly variable and can produce largely different colors depending on a multitude of factors such as different staining times, the variable concentration and pH of the staining solutions [12] or the stain suppliers. In the review article *the haematoxylin and eosin stain in anatomic pathology* [18] Mark R. Wick presents some of the specific problems that can occur during staining that cause variability in the quality of the sections. The irregular staining of the sections, a poor definition between the nuclei and the cytoplasm, an over- or understaining with either of the stains or a blue-black precipitate in the stained sections are some features that contribute to a low quality section.

The tissue samples that are analyzed in this project are sampled using a 5mm punch biopsy from the oral mucosa which is the mucous membrane of the inside of the mouth. The sample is then fixed in a paraformaldehyde solution to minimize the breakdown of the tissue structure before being dehydrated and embedded in paraffin wax. The paraffin embedded samples are then sliced into thin sections, placed on glass slides (often using a microtome [12]). These slices are then deparaffinised and rehydrated and the formerly described H&E staining is applied [5], [18]. After this the section is analyzed under a microscope or digitized with a scanner and analyzed on a computer screen. This digitization process can also introduce variabilities in the digitized tissue samples depending on the use of different digitization systems [12].

As these variabilities can have great consequences on the computational algorithms used to analyse the digitized tissue samples, image processing techniques can be used to normalize the samples and get a more consistent dataset [8]. This is

most commonly referred within digital histopathology as color management, see the pre-processing step in Fig. 1.

5) Oral mucosal tissue structure: The oral mucosa consists of two main layers, the epithelial layer and the lamina propria [19]. The epithelial layer is the outer-most layer and is formed by epithelial cells (Epith.). The lamina propria consists of multiple layers, the papillary layer and the underlying reticular layer [20]. They both contain fibroblast cells that produce collagen fibers. In the lower layer the cells are more spread out with thicker regions of collagen [5], [20]. Fibroblast cells (Fibr.) are present throughout lamina propria. Endothelial cells (Endo.) are lining vascular channels throughout lamina propria [20].

Lymphocytes are immune cells that appear in inflamed areas and are therefore not very present in healthy tissue. Both Inflammatory cells (Infl.) and Lymphocyte (Lymph.) are present in unhealthy tisse in areas of acute and chronic inflammation [20]. A large aggregation of lymphocytes is a sign of an active disease.

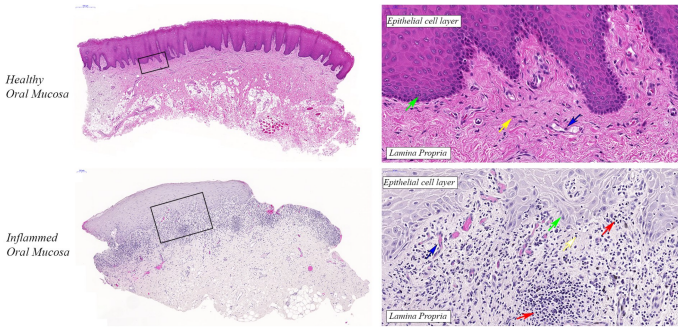


Fig. 2. Two WSIs of a healthy oral mucosa (top) and an inflamed oral mucosa (bottom). (Image provided by R. Sugars for Masters Thesis [5])

In Fig. 2 the black boxes locate the magnified areas on the WSIs. The arrows point to different cell types: Infl, (red), Epith. (green), Fibr. (yellow) and Endo. (blue).

Since some cells appear more frequently than others in the tissue, imbalance in the amount of labeled cells of different types is an inherent characteristic of histological datasets. As an example, Epith. cells appear much more in the tissue than Infl. cells since Epith. cells form the tissue structure while Infl. cells appear only in inflamed areas.

B. Color normalization

1) General: The term color normalization (CN) encompasses methods used to alter the color distribution of an image to fit certain needs. This can be achieved with many different methods such as histogram specification or generative adverserial networks [21], [22]. Within digital histopathology they can be applied as a pre-processing step to transform the tissue samples with variable colors to a common color space [2]. The aim of the CN algorithms in digital histopathology is to reduce the color variability in the dataset introduced by the staining. Since the color in digital histopathology images comes from staining, CN in digital histopathology is also commonly referred to as stain normalization.

2) Warwick toolbox: The CN methods in the Warwick toolbox [23] require a *target image* that defines the pursued color distribution for the methods to replicate on the input image.

The first type of CN method used is a color transfer method. In this method, the original RGB image is transformed to the perception-based $l\alpha\beta$ color space [21] and the mean and the variance is matched to the one from the target image. The method is implemented in the toolbox according to Reinhard et al. [21].

The second type of CN method uses a stain deconvolution method, introduced by Macenko et al. [15], it relies on a stain vector that represents the proportion of each wavelength absorbed and it characterises the stain present on the image. The challenge of stain deconvolution, however, is robustly estimating the stain vectors V, which should be done adaptively for each image [8]. This was further improved by Khan et al. [24] which is the state-of-the-art filter in this toolbox.

The Reinhard method was initially designed for general color transfer between images, while both Macenko and Khan are designed specifically for CN of digital histopathology images.

C. Machine learning

1) General: The use of Artificial Intelligence (AI), Machine learning (ML) and Deep learning (DL) has been called the fourth industrial revolution due to the fact that AI methods, tools and vocabulary are used in many fields to systematize and automate problems [25]. AI tools have become state-of-the-art in numerous medical applications by identification, quantification and classification of patterns in medical images to support practitioners [26], [27]. These developments make it possible to standardise and automate manual and subjective tasks, leading to more effective and efficient patient care [28].

2) Machine and deep learning models & the classification problem: The goal of ML models is to develop automated methods to detect and uncover patterns in data to make better predictions. This makes it similar to statistics, it differs primarily in its emphasis and terminology [29]. As Goodfellow et. al puts it "Machine learning is essentially a form of applied statistics with increased emphasis on the use of computers to statistically estimate complicated functions" [30]. This quote refers the *Universal approximation Theorem* presented in 1989 by Kurt Hornik [31] which states that a large enough ML model can estimate any complex function arbitrarily accurately. However, even though a ML model theoretically could approximate any function, that is far from the truth in practice. Usually this means that our ML-algorithm might not find the true value for our internal parameters, or it finds a wrong function [30].

Moreover, a DL model is a large ML model which deals with a vast number of parameters, in some cases even in the order of $10^6 - 10^7$ as for example with Google's EfficentNet [32]. This allows the algorithm to learn highly complicated patterns and requires a large amount of data [30].

ML methods can employ different types of learning: supervised, unsupervised and reinforcement learning. Supervised

learning tries to map inputs x to outputs y, given a training set $D = \{x_i, y_i\}_{i=1}^{N}$, where N is the number of data points. Each entry x_i is referred to as "features". This means estimating a function where a training set D of correctly identified observations is at your disposal [33]. Unsupervised learning however, tries to identify "interesting patterns" in data given no labels y_i, which make these problems less well-defined [29]. Supervised models address different types of classification problems, the most common are binary, multi-class and multi-labeled [34]. For multi-class problems, you have a training set D of different classes and our model predicts one of these classes for all data points in an unseen but similar dataset [35].

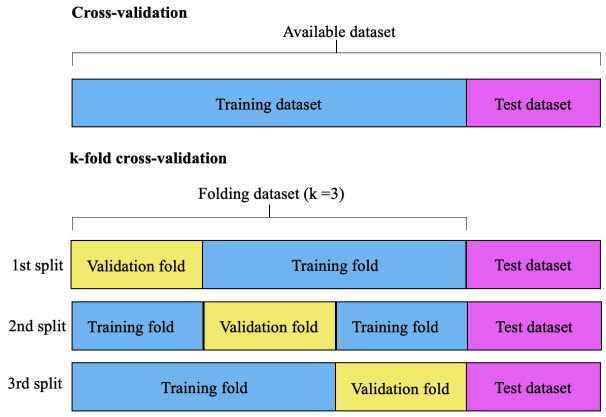


Fig. 3. A visualisation of the differences between test, validation and training set.

For ML models it is important to distinguish between training, validation and test data which is depicted in Fig. 3. Starting with a large data pool of available data, a portion of the data is taken out and marked as test data. The test data is used after training to understand how well the learned patterns generalize and is not used during training. The remaining data can then be split into training and validation. The training data is used to fit the internal parameters of the model, while the validation data is there to estimate generalization errors during training, optimize hyperparameters and to compare with the test data performance after training. This is called cross-validation. However, when a dataset is too small k-fold cross-validation is used instead, where the available dataset is partitioned in k splits randomly. If this is done by keeping the class distribution it is called stratified k-fold cross validation. By doing this we train k models to minimize the bias [30], [36].

All supervised ML algorithms contain essentially four properties: a model, internal (inside model) parameters and external parameters (also called hyperparameters), a notion of penalizing bad predictions per iterations (cost functions) and an optimization for minimization direction. Internal model parameters are specific for each model, for example a Neural Network model contains weight matrices W and biases b and a simple linear model has parameters slope k and m being y-intercept [30]. The hyperparameters are used to control the updating process of the internal parameters [30]. A cost (or loss) function is a real-valued function of the training and validation data. The cost function gives a numerical score where by convention a lower numerical score is better. Cost functions can be as exotic as cross-entropy loss or as simple as Mean Squared Error (MSE), known from undergraduate statistics class for linear regression models [37]. Something a statistician might call model fitting of internal parameters an ML researcher would call "learning" [38]. The training is, ideally, driving this numerical score towards gradually lower scores, so that the model learns. During the training process some model parameters grow out of scale, an activation function "restricts the values within an acceptable range" [39]. Its selection of activation function is dependent of the model and where in the model it is used. Common activation functions are Sigmoid, ReLU, ELU [39], [40]. Lastly, the cost function is minimized iteratively with an optimization algorithm. Since 1989 one of the main optimization algorithms has been gradient-based back-propagation [41], in short Back-prop. Backprop is essentially a numerical computation of the chain rule to compute the partial derivatives with respect to all internal parameters [30], [40].

Two common combinations of cost function and optimizer are Stochastic Gradient Descent (SGD) and Cross-Entropy Loss (CEL) mentioned above. CEL is the negative log-likelihood of our training labels, model parameters and input variable and has been found to lead to faster learning and improved generalization [30].

SGD is a common less computationally expensive solution in gradient-learning, to minimize a loss function L. When using SGD the gradient is estimated using a sample size n. One version of SGD uses two hyperparameters: learning rate l and momentum m. The pseudocode is presented in Algorithm 1. Momentum aims primarily to handle variances in the stochastic gradient due to our sampling method [30]. Learning rate is simply determining the size of the step. Most advanced learning algorithms adapt this parameter throughout the learning phase, for example by a decreasing scheme under the assumption that only incremental changes are needed later in the process when the minima is reached [30]. SGD is usually calculated on $32 - 515$ datapoints, since fewer datapoints tend to ease the learning process for DL models [42].

Algorithm 1: SGD with momentum [30]

Require: Learning rate l, momentum parameter m
Require: Initial parameter θ, inital velocity v
while *stopping criterion not met* **do**
  Sample a minibatch of m examples form the training set $x^{(1)}, ..., x^{(n)}$ with corresponding targets $y^{(i)}$
  Compute gradient estimate:
  $g \leftarrow \frac{1}{n} \nabla_\theta \sum_i L(f(x^{(i)}; \theta), y^{(i)})$
  Compute velocity update: $v \leftarrow mv - lg$
  Apply update: $\theta \leftarrow \theta + v$
end

3) Image classification models & Convolutional Neural Networks & EfficientNet: Image classification models form a subgroup of classification models that classify images. Among different approaches, Convolutional neural networks

(CNNs), a type of DL model, have become standard in image classification tasks and have achieved state-of-the-art results [43]. These neural networks use parameterized convolution kernels that preserve some of the spatial characteristics of the classified images [44].

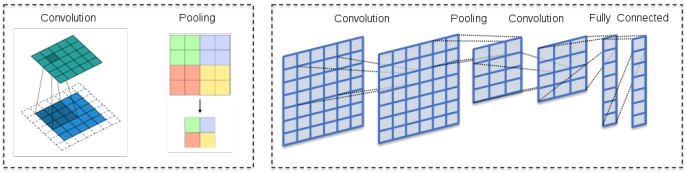


Fig. 4. Visual representation of convolution and pooling layers [45]

The convolution operation is done by sliding the kernel over the image and calculating the scalar product between the kernel and a specific part of the image. The kernel size defines the size of the image part to be scalar multiplied with the kernel. A convolutional layer of a CNN is made up of several of these kernels, where the conceptual idea for the different kernels is to learn local spacial features of the image [46]. To reduce the size of the input, pooling is then applied to the resulting convolutions. These two operations are visually represented in Fig. 4. A kernel in a CNN is a matrix randomly initialized which tries to abstract different features from the image. A feature is a specific representation of the image with some underlying pattern that seems to be evident by the algorithm. However, it is debatable what those features actually are representing for a human. Older versions of ML algorithms used to have hand-crafted features created by domain experts, for example edge detectors [1].

The number of parameters involved in these CNNs have since the success of AlexNet in 2012 [47] increased drastically to improve the accuracy of the models. The number of parameters is a compromise as too many parameters lead to a computationally expensive model and too few may compromise the results of the model. In the article *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks* [32] Tan and Le introduce an effective compound scaling method that reaches state-of-the-art accuracy and can scale to any target constraints. Different scales of the EfficientNet model are introduced named B0 to B7. The architecture of the model with the least parameters, EfficientNet-B0 is presented in Tab. I where MBConv stands for mobile inverted residual bottleneck blocks, which have shown to make the process more efficient [32], [48].

A common practice in DL and specifically in image classification is to utilize already trained models on large community (open sourced) hand-labeled datasets, for example ImageNet [49]. Transfer learning means using some parts of a pre-trained model and then training it for a new but similar task [30].

4) Limited dataset & class imbalance & data augmentation & overfitting: The models and learning algorithms used today are nearly identical, at least conceptually, to those used 20-30 years ago. The difference is that the availability of large amounts of data have reduced the level of competence needed for the user. The performance of a DL model is highly

TABLE I
EFFICIENTNET-B0 ARCHITECTURE

Stage	Operator	Resolution	Channels	Layers
1	Conv3x3	224 × 224	31	1
2	MBConv1, k3x3	112 × 112	16	1
3	MBConv6, k3x3	112 × 112	24	2
4	MBConv6, k5x5	56 × 56	40	2
5	MBConv6, k3x3	28 × 28	80	3
6	MBConv6, k5x5	14 × 14	112	3
7	MBConv6, k5x5	14 × 14	192	4
8	MBConv6, k3x3	7 × 7	320	1
9	Conv1x1 & Pooling & FC	7 × 7	1280	1

dependent on the amount of data available during training [30]. When large amounts of data is not available the performance generalizability of the model becomes harder to adjust, especially if the dataset has an inherent class imbalance for representative examples. Models with low generalizability are overfitted and perform well only on the training data, which can be seen when comparing the validation and test accuracy since validation is a subset of the training set [44]. This means that when a model has not generalized well it has not learned broad enough features from the training data to be able to interpret variations in the test dataset. Models trained on imbalanced datasets show worse performance than those trained on balanced datasets especially for classification problems [50].

One technique to handle class imbalance is to use oversampling, where we use multiple versions of the same data from the same class. By doing this we can balance out our dataset at a cost of increased risk for overfitting [50].

Moreover, there exists a lot of different label-preserving data augmentation techniques to minimize overfitting. A common method is random cropping, which involves resizing the sample and interpolating the new pixel values, to later randomly crop to a chosen size. Another method is noise injection, which adds imperceptible perturbations to the images. A specific noise is adversarial noise, this noise is specifically created to make the model make wrong predictions [51].

Regularization techniques minimize overfitting for limited datasets by using a weight decay that adds a regularization term to the cost function which supports the optimization algorithm getting closer to a minimum [30].

5) Ensemble learning: An ensemble learning method combines multiple model predictions into one prediction. The multiple models are trained independently and each of them are given a vote in evaluation of the test data. This has been found to increase generalization, minimize error in predictions and decrease variance of predictions [26], [30]. The most commonly used ensemble learning methods are majority voting, probability score averaging (PSA) and stacking ensemble [26].

6) Model evaluation methods: To evaluate empirical notions of accuracy and performance of a classification model different measures can be used to help assess the models ability to predict the classes of the unseen data.

To evaluate a classification model for a dataset with class imbalance it is standard to use Precision, Recall and F1-score

per n classes. To scale these measures to tell us something about the entire model accuracy, macro and weighted measures are used, see Eq. 1 -14 below. To understand these measurements it's more intuitive, and eases the notation, to start with the classification (confusion) matrix. A confusion matrix is a visualisation tool where the entries are basis for all accuracy metric calculations in classification problems arising in multiple fields from computer vision to natural language processing [35]. The matrix, see Tab. II, has entries $x_{i,j}$, column j being actual label and row i being model prediction, opposite conventions exists in some literature. Each entry is the number of datapoints with the predicted and actual label of that particular row and column.

TABLE II
GENERALIZED CLASSIFICATION (CONFUSION) MATRIX

	Class 1	Class 2	Class 3	...	Class N
Class 1	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	...	$x_{1,N}$
Class 2	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	...	$x_{i,N}$
Class 3	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	...	$x_{i,N}$
⋮	⋮	⋮	⋮		⋮
Class N	$x_{N,1}$	$x_{N,2}$	$x_{N,3}$	...	$x_{N,N}$

For the binary classification problem our NxN dimension confusion matrix becomes a 2x2 matrix, see Tab. III. Each entry has an associated name, where $x_{1,1} :=$ True Positive Count (TP), $x_{1,2} :=$ False Positive Count (FP - Type 1 error), $x_{2,1} :=$ False Negative Count (FN - Type 2 error) and $x_{2,2} :=$ True Negative Count (TN) [7], [52].

TABLE III
BINARY CLASSIFICATION (CONFUSION) MATRIX

	Class 1	Class 2
Class 1	TP	FP
Class 2	FN	TN

We can now define our metrics for the multi-class and binary classification problem. For the binary classification problem, with only two classes the common measures are:

$$Accuracy := \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

$$Precision := \frac{TP}{TP + FP} \tag{2}$$

$$Recall := \frac{TP}{TP + FN} \tag{3}$$

$$F_1 - score := 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{4}$$

For each Eq.1 - 4 one can investigate each metrics purpose or "Evaluation focus", inspired by Sokolova (2009) [34]. This will give an intuition for each measure before we generalize to datasets with more classes than two. $Accuracy$ evaluates how well the model predicted the two classes, or the overall effectiveness of the model. $Precision$ is the fraction of the predicted positives which were actually positive. $Recall$ is the fraction of how many of the actual positives where predicted as such. When evaluating performance on an unbalanced datasets accuracy is a poor performance metric for characterizing and $Precision$ and $Recall$ give a better representation of model performance [30], [33]. In specific applications $Recall$ or $Precision$ could be especially important. For example, in digital pathology $Recall$ is important since false negatives could mean missing to diagnose a certain disease. Lastly, $F_1 - Score$ is a harmonic mean of $Precision$ and $Recall$ [52], which penalizes a spread amongst $Precision$ and $Recall$ in a better way than an arithmetic mean. For example, $Precision = 0.1$ and $Recall = 0.9$, would give $\overline{x} = 0.5$ while $F_1 - Score = 0.18$ which gives us a better representation of this poorly performing model.

For the multi-class classification problem there are N classes or states, which means that Eq. 1 - 4 has to be generalised for more than two classes. We will use the introduced notation from Tab. II.

$$Accuracy := \frac{\sum_{i=1}^{N} x_{i,i}}{\sum_{i=1}^{N}\sum_{j=1}^{N} x_{i,j}} \tag{5}$$

$$Precision_n := \frac{x_{n,n}}{\sum_{i=1}^{N} x_{n,i}} \tag{6}$$

$$Recall_n := \frac{x_{n,n}}{\sum_{i=1}^{N} x_{i,n}} \tag{7}$$

$$F_1 - Score_n := 2 \cdot \frac{Precision_n \cdot Recall_n}{Precision_n + Recall_n} \tag{8}$$

Note that measures above for $Precision$, $Recall$ and $F_1 - Score$ are per class n of N classes, denoted $Precision_n$, $Recall_n$ and $F_1 - Score_n$. A $Precision$, $Recall$ and $F_1 - Score$ measure for the entire model could also be of interest to give a notion of average $Precision$, $Recall$ and $F_1 - Score$. Most common approaches here is to combine each measurement across classes with an normal arithmetic average, also called macro average, here denoted ^{MA}, see Eq. 9-11.

$$Precision^{MA} = \frac{1}{N}\sum_{n=1}^{N} Precision_n \tag{9}$$

$$Recall^{MA} = \frac{1}{N}\sum_{n=1}^{N} Recall_n \tag{10}$$

$$F_1 - Score := 2 \cdot \frac{Precision^{MA} \cdot Recall^{MA}}{Precision^{MA} + Recall^{MA}} \tag{11}$$

Class imbalance can be accounted for by multiplying each measurement with its associate weight w_i (number of class appearance divided by total number of points), also called weighted macro average, here denoted ^{WMA} [34], see Eq. 12- 14.

$$Precision^{WMA} = \frac{1}{N}\sum_{n=1}^{N} w_n \cdot Precision_n \tag{12}$$

$$Recall^{WMA} = \frac{1}{N}\sum_{n=1}^{N} w_n \cdot Recall_n \tag{13}$$

$$F_1 - Score := 2 \cdot \frac{Precision^{WMA} \cdot Recall^{WMA}}{Precision^{WMA} + Recall^{WMA}} \quad (14)$$

7) Limits of DL algorithms and intrinsic variability: Much of the research in DL can be explained by the *no free lunch theorems* which state that each class of optimization problems need specific solutions [53]. Just because an DL algorithm works on a specific sub-problem and dataset it does not mean it will work on a similar setups, meaning there is no superior DL algorithm for all uses [30]. Another issue with DL models is variability. One of the most cited papers within this area, written by Dietterich [54], concluded in 1998 that there are multiple random sources causing the variability. These sources include random variation in data selection, internal randomness in common algorithms and random classification errors. This causes challenges for reproducibility within the academic community.

The training of the large models used in DL requires large computational resources and time allocation. Cutting edge research in the field often needs a high performance computer cluster for effective training [40].

Concerns have been raised about the unexplainable nature of decisions made by DL algorithms and a need for them to be interpretable by humans. This is called explainable AI and deals with methods that "enable causality, explanatory ability, and interpretive ability of the prediction results of the model" [55], [56], which may be important in some applications such as diagnostic tasks within the medical field.

D. Related work

In this paragraph relevant research will be presented as a basis for the discussion of this paper and a highlight of their findings. Pontalba et al. [8] used the CNN models named CNN3 [57] and UNET3 [58] for a segmentation task based on three different H&E stained datasets TCGA [57], TNBC [59] and SMH (not public) [8]. Stain normalization was used as a pre-processing step with the Warwick toolbox [23]. The papers showed that the filters introduced variability and used the ensemble method PSA. The paper only presented validation metrics with dice similarity coefficient (DSC) [8] as the main performance metric. The performance varied for the TCGA and TNBC dataset where the ensemble learner performed better than the individual filters with the DSC metric.

Estreen [7] used the EfficientNet model [32] on the H&E-stained KI dataset. However, Estreen did not use the same training/validation/test composition as this report. The report also displayed validation metrics of accuracy $\sim 92\%$ as the main result. Within the same research group Brynjarsson [5] used shallow neural networks on the KI dataset, VGG16, RCCNet & Softmax. The data composition was slightly different to this report. The main presented results were in terms of validation accuracy where the architectures performed as following: VGG16: $\sim 85\%$, RCCNet: $\sim 88\%$ & Softmax: $\sim 90\%$.

Following will be an exposition of previously reported results from object detection models from similar fields. Chouhan et al. [60] showed an increase of ~ 2 percentage points on accuracy from their best performing model on a similar task using a majority voting technique on a classification task. Shorfuzzman et al. [61], similarly to Chouhan et al. showed an increase of ~ 1 percentage point. Lakhani et al. [62] showed an increase of ~ 1 percentage point on AUC (Area under the ROC Curve) from their best performing detection model with PSA. Hooda et al [63] showed a ~ 4 percentage points increase on a similiar task as Lukani et al. presented. Hinton et al. [64] reported interesting results on a speech recognition model using ensemble learners. The use of ten models for ensembles only increased the accuracy ~ 2 percentage points.

Islam et al. [65] showed that ensemble learning models experience a diminishing return on investment at around 5-10 models. We cannot continue to add models and expect the performance to continuously increase. However, the robustness of the model improves as more models are added. PSA is also expected to perform better than majority voting.

III. Methods

The process pipeline consisted of four different parts that we are accounted for in order here. The process setup is visualized in Fig. 5 and each component is addressed below.

A. KI Dataset

The dataset consists of partially labeled 2000 x 2000 pixel WSI of tissue samples from six patients. The tissues have a great variety of histological grade (G0-G4) [14] and quality of the H&E staining.

The images were labeled by pathologists at KI with an open source tool called LabelIMG [66]. The tool generates markup-files (.xml) with each labeled object having a name (cell type) and location (bounding box). Four different types of cells were labeled: Inflammatory (Infl.), Lymphocyte (Lymph.), Fibroblast and Endothelial (Fibr./Endo.) and Epithelial (Epith.). See Tab. IV-VII for their specific occurrences in the dataset, Fig. 6 for a visualization of their location in the WSI and Fig. 7 for a visualization of the individual cropped cell images.

TABLE IV
Training set

Grading	Patient ID	Images	Image labels
G0	P20	21	13 121
G3	P9	7	1 261
G4	P19	6	1 234
Tot.		34	15 616

TABLE V
Training set per cell type

Grading	Patient ID	Infl.	Lymph.	Fibr./Endo.	Epith.
G0	P20	420	935	3 745	8 021
G3	P9	148	404	284	425
G4	P19	147	628	199	260
Tot.		715	1967	4 228	8 706

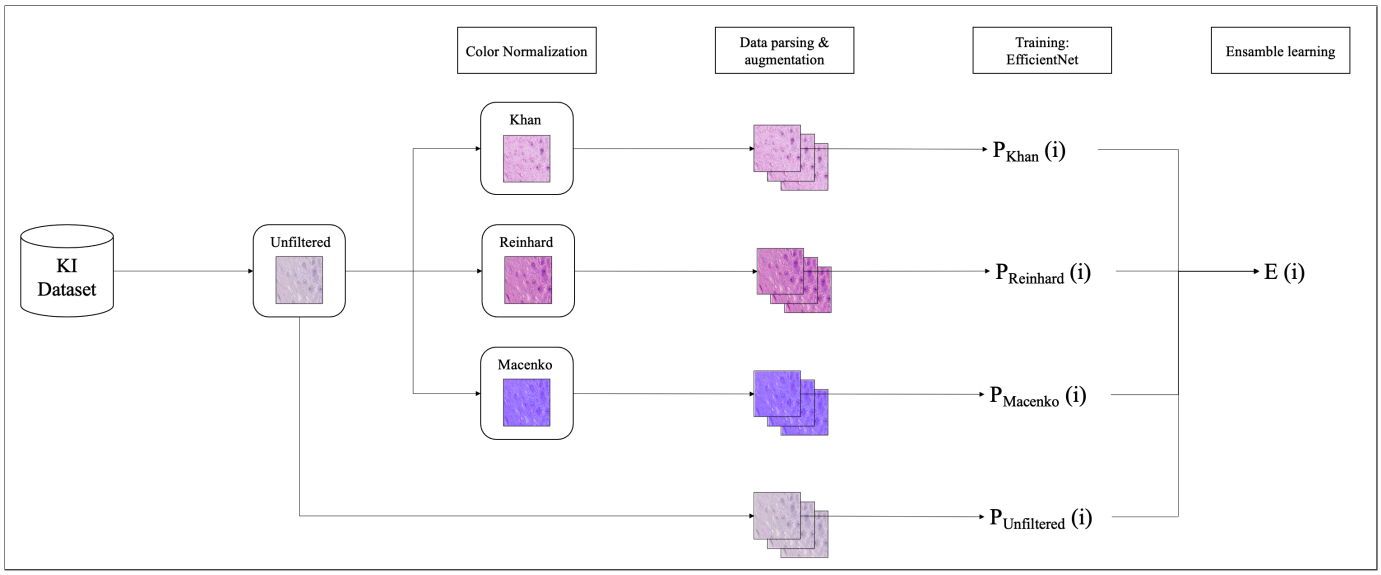


Fig. 5. Visualization of project pipeline

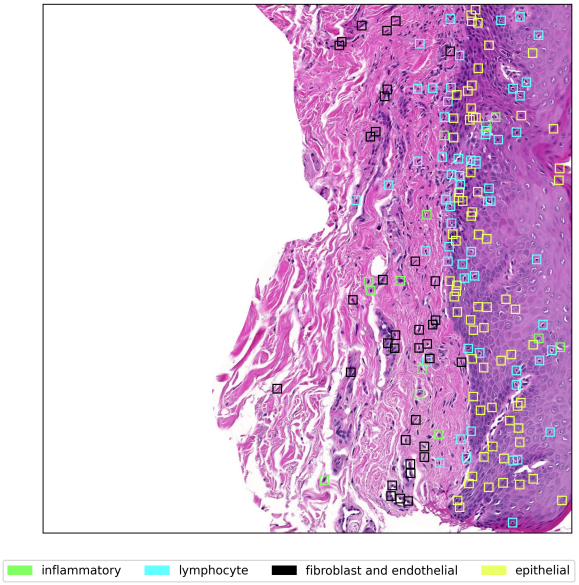


Fig. 6. Visualization of location on type of the annotated cells in the 2000x2000 slide P9_4_1

TABLE VI
TESTING SET

Grading	Patient ID	Images	Image labels
G0	N10	5	2 094
G3	P13	4	2 004
G4	P28	4	7 220
Tot.		13	11 318

B. Color Normalization

All images were filtered using three publicly available CN algorithms based on *Stain Normalization Toolbox* provided by the Department of Computer Science of the University of Warwick [23]. These three filters are referred to as Macenko [15], Reinhard [21] and Khan [24]. The image P9_4_1 from the KI Dataset was used as a reference image for all three CN algorithms. This specific image was chosen because it showed good staining quality. Every single image in the dataset was then filtered with the three filters and saved as four different

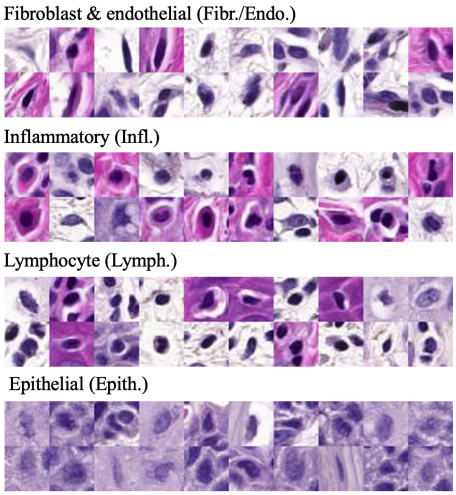


Fig. 7. Visualization of different cell types in our dataset. The cells are cropped from images P20_5_1, P9_3_1 and P19_2_1.

TABLE VII
TESTING SET PER CELL TYPE

Grading	Patient ID	Infl.	Lymph.	Fibr./Endo.	Epith.
G0	N10	79	59	634	1 322
G3	P13	333	278	787	606
G4	P28	729	1 987	1 775	2 729
Tot.		1 141	2 324	3 196	4 657

image files (one unfiltered, and one for each of the three filters).

C. Data parsing & augmentation

The 2000x2000 WSIs are cropped to 32x32 images around the center of the labeled cells, creating a separate image per label (see Tab. IV & VI and Fig. 7). Note that unlabeled cells were not taken into account.

All parsed images were Gaussian normalized to minimize the impact of intensity and contrast variability, and transformed pixel means above 0.95 were considered white and ignored [7].

A limited dataset is prone to issues with overfitting, less generalizability and biased models. To address these first two problems, five data augmentation methods were used on the training and validation set. The first method is random cropping and involved resizing the images from 32x32 to 246x246, the new pixels are interpolated using bicubic interpolation [67], [68]. After the resizing to 246x246, the images are cropped around a randomly chosen location to 224x224. In the second method, Gaussian Noise was added to all pixel values ($\mu = 0 \,\&\, \sigma = 0.1$). Thirdly, additional training samples were created by injecting adversarial noise (AdvProp). Fourthly, a regularization technique was used together with the chosen optimizer, which set weight decay to 0.00005. Lastly, also due to the selection of small-batch regime optimizer, the batch size was set to 32. To address the bias associated with a limited dataset, stratified k-fold cross validation techniques were used with $k = 5$. Because of the imbalanced dataset, oversampling techniques were used to equalise the magnitudes over our four class (cell) types. This oversampling involved augmenting the dataset, see Tab. VIII.

TABLE VIII
TRAINING SET PER CELL TYPE - BEFORE AND AFTER OVERSAMPLE

Total cell type count	Infl.	Lymph.	Fibr./Endo.	Epith.
Before oversample	715	1 967	4 228	8 706
After oversample	6 435	9 835	8 495	10 706
Scaling factor	900%	500%	200%	123%

D. EfficientNet: Cell classification model

The cell classification model is based on Google's *State-of-The-Art* Convolution Neural Network EfficientNet [32] through an early January 2020 open source implementation [69], using the model architecture with least parameters: EfficientNet-B0. The implementation is using Facebook's open source deep learning framework called PyTorch [70]. Weights were updated under training and cross entropy loss was selected as the cost function. To ease the computational time Stochastic Gradient Descent (SGD) was used to estimate the actual minimization direction. The learning rate, the cost function multiplier, was set to 1%. A learning rate schedule was used by decaying it 3% every 2.4 epochs (iterations), to converge to the minimum in a smoother manner. To accelerate the training and to get better performance with fewer epochs, momentum was set to 0.9. Transfer learning was used by loading a model pretrained on the ImageNet dataset [7].

E. Training and implementation

For each of the 5 validation splits 100 iterations were used and the computation time for each trained model with selected hyperparameters was in the magnitude of 8 hours. Over the 100 epochs the model with the highest validation Top1-accuracy was chosen. Computations were made using NVIDIA V100 and T4 nodes on a Swedish National Infrastructure for Computing (SNIC) resource named Alvis [71] through the grant agreement no. 2020/33-67. The entire project is documented and available at github [72].

F. Ensemble learning

Two ensemble learning regimes were implemented based on all five trained models (k=5, see above) for each four datasets with six iterations, giving us a total of 120 models to combine. The first algorithm was based on a naive approach where each models top1-class prediction per image was combined via a majority voting technique. This means that if three models predicted Lymph. and one Infl., Lymph. was predicted. We dealt with equal predictions by uniformly randomizing the prediction, see Algorithm 2.

In the following pseudocode explanation, N refers to the number of different filters used, including the unfiltered version, in our case $N = 4$. M refers to number of data points in the dataset, in our case the testing set has $M = 11\,318$.

Algorithm 2: Naive majority voting ensembling

Input : N vectors y_j where $y_j(i)$ is the Top1 predicted class of the i:th data point
Output: the vector e where $e(i)$ is the ensemble predicted class of the i:th data point
for $i \leftarrow 1$ **to** M **do**
  Assign to $e(i)$ the most common value $y_j(i)$ for j between 1 and N;
  If two or four values appear the same amount of times, we randomly choose one of the two or four
end

The second more restrictive and representative regime PSA was inspired by the Pontalba paper [8], [26], see Algorithm 2. A softmax activation function was applied to the output layer, giving us four probability vectors (P_i) for each image and model. These were added up, and scaled down by model count, see Eq. 15.

$$E_i = \frac{1}{N} \sum_{filter} P_i \tag{15}$$

G. Model evaluation

To evaluate the model precision, recall and F1-score were used on the top1 prediction for each cell type, including an average for all cell types, with and without weights. Accuracy refers to Top-1 accuracy which uses the Top-1 class, the class with the highest predicted probability. The Top-1 class is also used to calculate precision and recall. Confusion matrices were added to strengthen the analysis per class. Lastly, loss and

Algorithm 3: PSA ensembling

Input : N matrices p_j where $p_j(i)$ is the vector with the predicted class probabilities of the i:th data point

Output: the vector e where $e(i)$ is the ensemble predicted class of the i:th data point

for $i \leftarrow 1$ **to** M **do**

  Assign to V the sum of $p_j(i)$ for j between 1 and N;

  Scale down the V with N for each element.

  Assign to $e(i)$ the top1 predicted class of V

end

accuracy curves were used to study the learning process during the training phase.

IV. Results

A. Color Normalization

The first step where results could be seen was during the pre-processing step. Depicted in Fig. 8 we show some representative parts of the color normalized dataset over a variety of histological severity and oral mucosa structures.

B. Classification Performance Metrics

The 120 trained models were all evaluated on the testing set filtered in the same manner as the training set. The selection of models for the ensemble method were arbitrary as long as they were unique and one from each filter. No model was used in the ensemble learning more than once. Presented are our findings with our independent variable being the filters and all other parameters were fixed during our experiments. The evaluation metrics from these classifications are presented in the Fig. 9 - 12.

In Fig. 9, the boxplots show the accuracy per filtered model type for the four differently filtered datasets and the two ensemble learning techniques. The second Fig. 10 gives additional insights with the metrics precision, recall and F1-score which was averaged in two different ways. Fig. 11 shows how all model types performed over our four cell types. In Fig. 12 the relationship per model type is presented. This figure aims to investigate if a certain filter helps the trained model to classify a certain type of cell.

To further visualize the results, the metrics of the best performing model in each model type is presented in Appendix A. The appendix includes a confusion matrix to provide a better understanding of the classification. To show the learning process over epochs the appendix also includes loss and accuracy curves.

V. Discussion

A. Color normalization

The quality of the CN results differed between filters but also between different files when the same filter was used, see Fig. 8. It is possible that for each filter, artifacts from CN could arise. Both the filters Khan and Reinhard seem to produce a result that mimics a high quality H&E staining. Both the above statements are similar to the results presented by Pontalba et al. [8]. However, the filter Macenko introduced artifacts for some image files that manifested as a very blue color that did not relate to the original image in any apparent way and as yellow and red patches on some parts of the image. Pontalba et al. [8] also referred to Macenko as showing ”inconsistent color mapping”. However the filter was kept to add variety in our results and to study its impact on model performance. According to Khan et al. [24] their filter is state of the art due mainly to three reasons: less introduction of artifacts, robustness and appropriateness for H&E staining. Khan et al. also claim that color deconvolution based methods, such as Macenko and Khan, are most appropriate for stain normalization since the ”chemical processes are largely independent for each stain, and color deconvolution separates out the effect of variation of each stain so it can be corrected independently”. Khan et al. [24] propose two reasons why Khan outperforms Macenko: ”1) Better, or more robust, deconvolution matrix estimation, and/or 2) a more appropriate mapping function.” Furthermore, Khan et al. [24] argue that the Reinhard filter is ”attractive in its simplicity but is based on the false assumption of unimodal color distribution in each channel” which is not true for dyes and stains.

Images that mimic high quality staining are expected to lead to better model training and classification. Conversely, inaccurate coloring and introduced artifacts are expected to negatively affect model training and cell classification.

B. Classification Performance Metrics

All filtered model types show an increased spread in accuracy values in relation to the non-filtered baseline model, see Fig. 9.

Even though Khan and Reinhard showed similar visual staining quality, the models trained on them performed vastly differently. Khan showed the highest accuracy and averaged metrics out of any of the individually filtered model types, see Fig. 9 and Fig. 10. On the other hand, Reinhard performed similarly to the unfiltered model type in accuracy and averaged metrics. The models trained on the Macenko filtered dataset had lower accuracy and averaged metrics than all other model types, see Fig. 9. The poor performance can perhaps be explained by the introduced blue artifacts of the Macenko filter. The artifacts might be due to the assumption of unimodal distribution presented in Ch. V-A but it needs further investigation for a definite conclusion to be drawn.

Both ensemble learners showed more robust accuracy values than any of the individually filtered models, see Fig. 9. The PSA ensemble learners outperformed the naive approach for accuracy and for the averaged metrics in Fig. 10. This was expected since PSA utilizes more information from each model. These results are consistent to what was presented in Ch. II-D. The ensemblers with 30 included models also showed very similar accuracy values to the best performing filter, Khan. This is interesting since the ensemble learners take into account the results from the poorly performing Macenko filter. This would indicate that further improvements to the

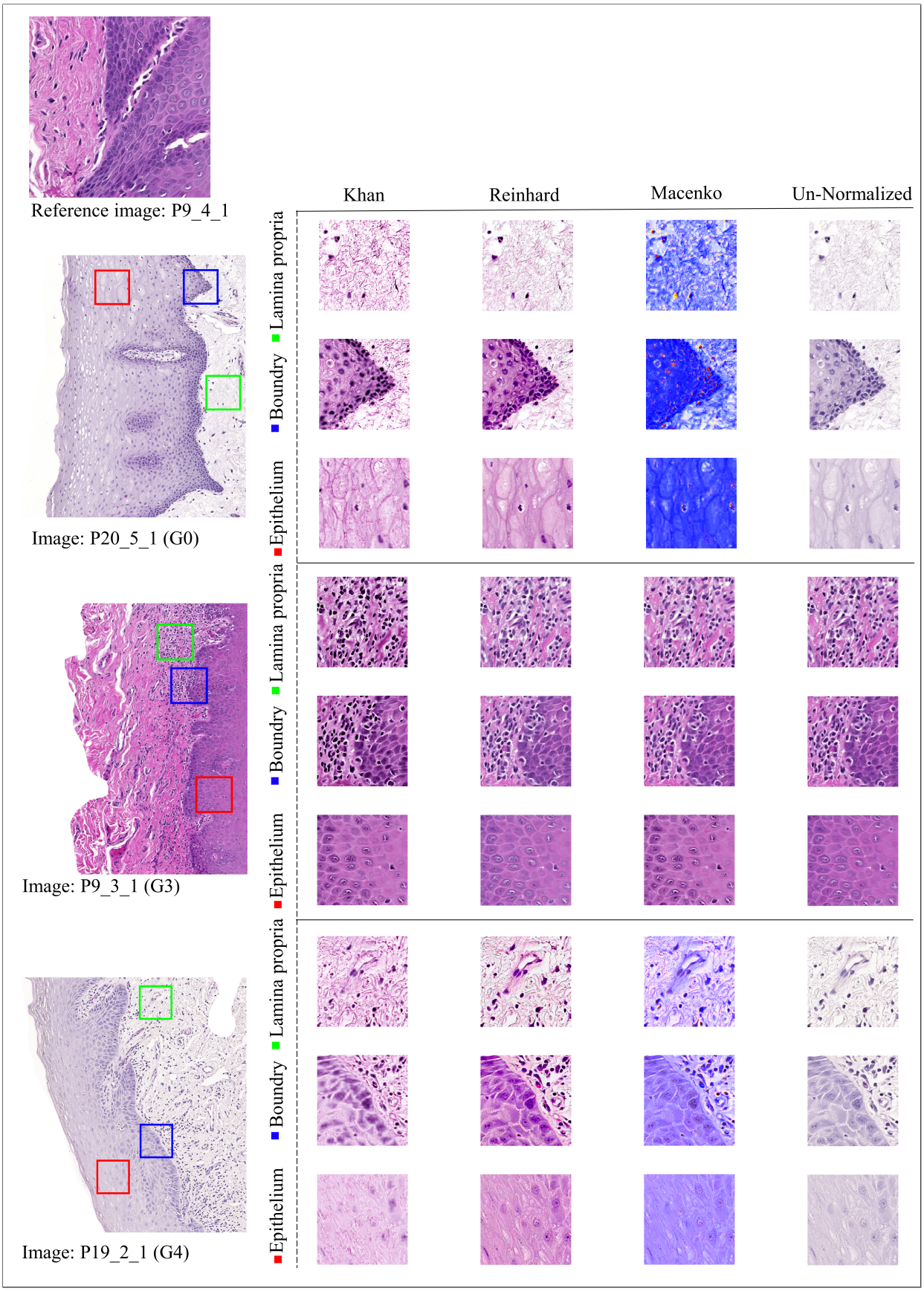


Fig. 8. CN filter results with varieties in histological grading (G0, G3 and G4) and oral mucusa structure: lamina propria (green), epithelium (Red) and the boundary (blue) in-between them. Meaning, for each image in the column "Un-Normalized" the three filtered results (Khan, Reinhard and Macenko) are depicted to the left with respect to a representative reference image.

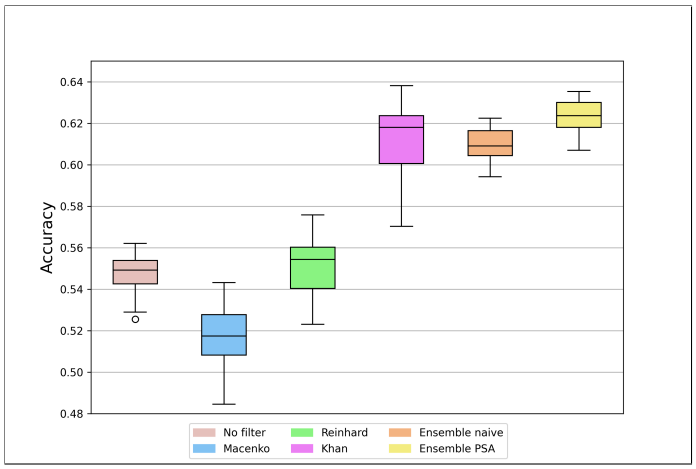


Fig. 9. Boxplot of accuracy results per filter and ensemble learning method. Evaluated on the test set, each box contains 30 data points (bincount).

performance could be made if models for the ensemblers are hand-picked and the Macenko filter is left out.

There is a clear trend for higher quality classification for cell types with more labels in the training dataset, see Tab VIII and Fig. 12. Specifically, for the inflammatory cell type all filtered datasets including non-filtered showed low quality classifications, see Fig. 11 and Fig. 12. This means that the used data augmentation techniques were not sufficient to prevent low quality classification for the rarest cell types.

One outlier from the general trend in filter performance was the classification of Lymphocytes where precision as expected was higher for Khan, but the recall was highest for Macenko. For classification of Epithelial cells the precision scores for Khan outperformed both ensemble learning techniques. Another trend shift can be seen for the inflammatory cell types where the ensemblers got worse metrics than the individual filters for recall, leaving the unfiltered dataset with the highest F1-score, see Fig. 12.

It is also clear that our model is highly overfitted when comparing the metrics from the validation and test set in Appendix A, where a 40 percentage points difference in accuracy values between validation and test datasets can be observed. The authors find it difficult to compare their results directly to Brynjarsson [5] who presents the validation metrics as the main result. However, the validation results in this report are in the same magnitude as both Estreen and Brynjarsson with a validation accuracy of $\sim 90\%$. The test accuracy presented by Estreen [7] is also similar to the test accuracy of this report, around $\sim 65\%$.

VI. Conclusions

For this project, three CN algorithms were used as a preprocessing step on the KI dataset to improve an existing EfficentNet CNN cell classification model. The CN algorithms were used to handle the color variability in the KI dataset. Performance was assessed by analysing the results from the individually trained models and by combining these results with ensemble learning techniques. Our conclusions are clear, stain normalization filters significantly impact classification performance. When we have a filter that introduces artifacts, such as the Macenko filter, the classification performance is worse than that of the unfiltered baseline. For filters with adequate staining qualities such as Khan, the performance is enhanced. Lastly, ensemble learning techniques have been shown to average out badly performing filters and giving us a robust performance, comparable to the best filter. We can conclude that the combination of well designed and selected CN methods and ensemble learning techniques boost performance for a cell classification model.

VII. Future work

Firstly, there are some issues in the existing pipeline that needs to be addressed. The oversample scheme is currently flawed and creates misleading validation metrics. The oversample process is performed on the training set before the training validation split. Therefore, some images can appear in both the training and the validation set multiple times. The chosen hyperparameters of the model need to aim for good performance on the test set rather than the validation set. Another main issue with the setup is our dataset that is still very limited making the class imbalance an especially difficult problem. There are shared and publicly available histopathological imaging datasets that are used in research [73] that could be used, however, none of these cover oral tissue samples specifically. It would therefore be of interest to expand the current dataset with more hand-labeled data possibly by collaborating with other institutes.

Ethics statement

The color normalization algorithms used in this paper are open source, and the KI dataset has been anonymized and does therefore not contain any information that could be linked back to any individual. Approval for the use of the image set of oral mucosal digitized histological slides for machine learning was granted by the Swedish Ethical Review Authority (Etikprövningsmyndigheten) Dnr: 2019-01259.

Appendix A
Scorecards per model

Acknowledgment

The computations and data handling were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) at C3SE Chalmers University of Technology partially funded by the Swedish Research Council through grant agreement no. 2020/33-67. Research funding was also provided by ALF Medicine and SOF Clinical Odontological Research Funding.

The authors would like to thank Karl Meinke and Rachael Sugars with team for their continued support and guidance during this project. The authors would also like to mention and thank Miritt Zisser at KTH Library for her support with the reference manager system Mendeley and our research process. Finally the authors thank their respective families for their continued encouragement and support during the process of this thesis. This accomplishment would not have been possible without the help of any of the aforementioned people.

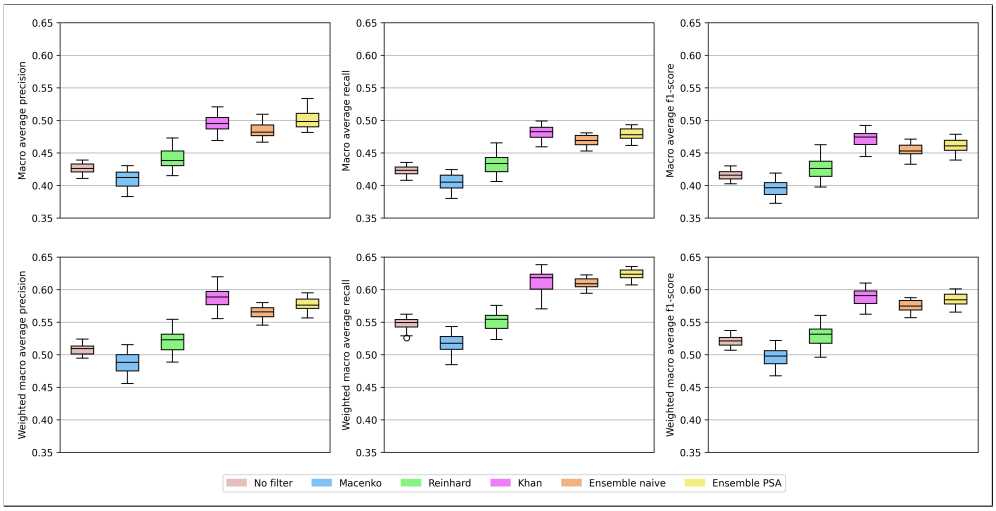


Fig. 10. Boxplot of combined (over all 4 cell types) precision, recall and F1-score values using macro average and weighted macro average. Evaluated on the test set, each box contains 30 data points (bincount).

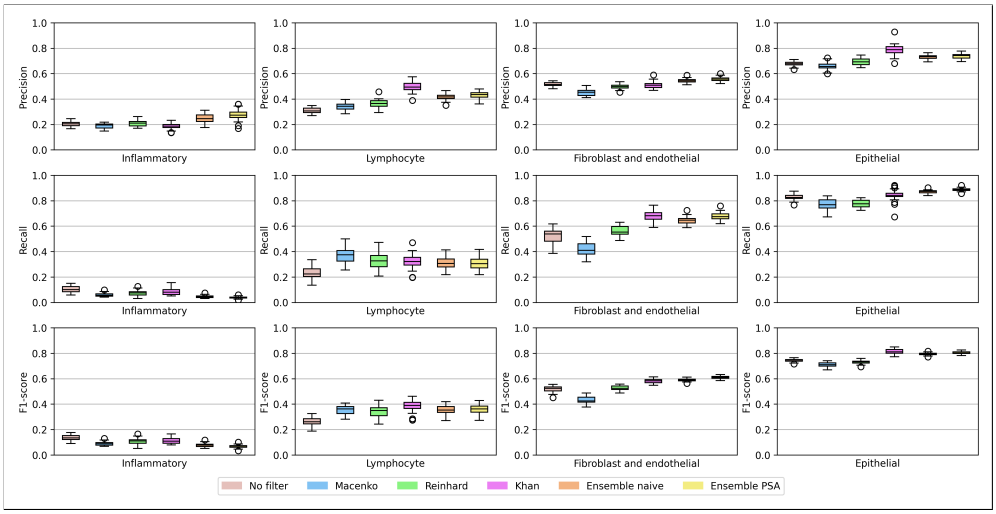


Fig. 11. Boxplot of precision, recall and F1-score per cell type. Evaluated on the test set, each box contains 30 data points (bincount).

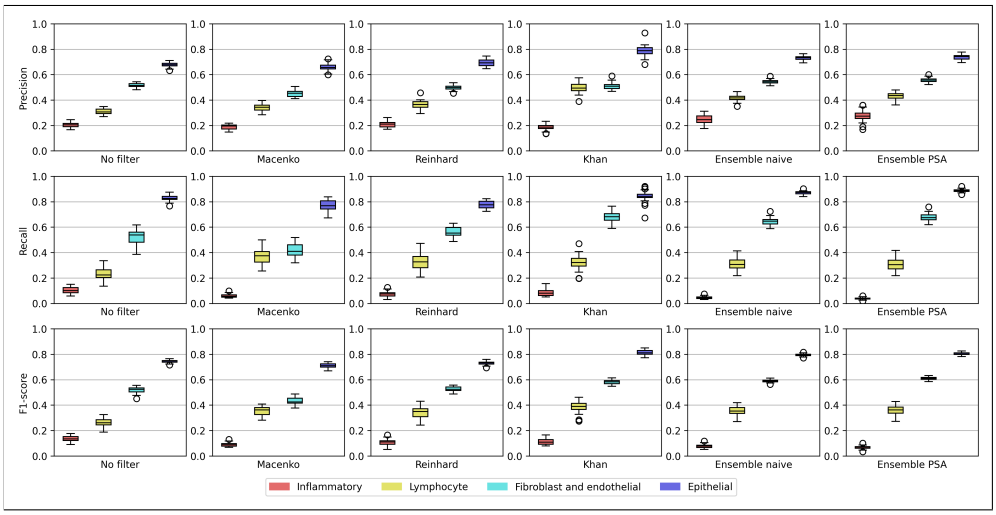


Fig. 12. Boxplot of precision, recall and f1-score per filter and ensemble learning method. Evaluated on the test set, each box contains 30 data points (bincount).

REFERENCES

[1] S. Deng, X. Zhang, W. Yan, E. I. Chang, Y. Fan, M. Lai, and Y. Xu, "Deep learning in digital pathology image analysis: a survey," *Frontiers of Medicine*, vol. 14, no. 4, pp. 470–487, Jul. 2020.

[2] B. Smith, M. Hermsen, E. Lesser, D. Ravichandar, and W. Kremers, "Developing image analysis pipelines of whole-slide images: Pre- and post-processing," *Journal of Clinical and Translational Science*, vol. 5, no. 1, pp. 1–11, Aug. 2021.

[3] S. Sayed, W. Cherniak, M. Lawler, S. Tan, W. Sadr, N. Wolf, S. Silkensen, N. Brand, L.-M. Looi, S. Pai, M. Wilson, D. Milner, J. Flanigan, and K. Fleming, "Improving pathology and laboratory medicine in low-income and middle-income countries: roadmap to solutions," *The Lancet*, vol. 391, Mar. 2018.

[4] M. Wilson, K. Fleming, M. Kuti, L.-M. Looi, N. Lago, and K. Ru, "Access to pathology and laboratory medicine services: a crucial gap," *The Lancet*, vol. 391, no. 10133, pp. 1927–1938, Mar. 2018.

[5] G. R. Brynjarsson, "Classifying nuclei in soft oral tissue slides," Master's thesis, KTH, Stockholm, Sweden, 2019.

[6] (2021, Apr.) Kebnekaise, swedish national infrastructure for computing. [Online]. Available: https://www.snic.se/resources/compute-resources/kebnekaise/

[7] T. Estreen, "Epithelial Layer Boundary Detection Using Graph Convolutional Networks for Digital Pathology," Master's thesis, [unpublished], KTH, Stockholm, Sweden, 2020.

[8] J. T. Pontalba, T. Gwynne-Timothy, E. David, K. Jakate, D. Androutsos, and A. Khademi, "Assessing the impact of color normalization in convolutional neural network-based nuclei segmentation frameworks," *Frontiers in Bioengineering and Biotechnology*, vol. 7, p. 300, 2019.

[9] A. Khademi, "Image analysis solutions for automatic scoring and grading of digital pathology images," *Canadian Journal of Pathology*, vol. 5, no. 2, pp. 51–55, Jun. 2013.

[10] N. Farahani, A. V. Parwani, and L. Pantanowitz, "Whole slide imaging in pathology: advantages, limitations, and emerging perspectives," *Pathology and Laboratory Medicine International*, vol. 7, pp. 23–33, Jun. 2015.

[11] J. Griffin and D. Treanor, "Digital pathology in clinical use: Where are we now and what is holding us back?" *Histopathology*, vol. 70, pp. 134–145, Jan. 2017.

[12] T. A. Azevedo Tosta, P. R. de Faria, L. A. Neves, and M. Z. do Nascimento, "Computational normalization of H&E-stained histological images: Progress, challenges and future potential," *Artificial Intelligence in Medicine*, vol. 95, pp. 118–132, Apr. 2019.

[13] H. Tizhoosh and L. Pantanowitz, "Artificial intelligence and digital pathology: Challenges and opportunities," *Journal of Pathology Informatics*, vol. 9, Nov. 2018.

[14] V. Tollemar, N. Tudzarovski, G. Warfvinge, N. Yarom, M. Remberger, R. Heymann, K. Garming Legert, and R. V. Sugars, "Histopathological Grading of Oral Mucosal Chronic Graft-versus-Host Disease: Large Cohort Analysis," *Biology of Blood and Marrow Transplantation*, vol. 26, no. 10, pp. 1971–1979, Oct. 2020.

[15] M. Macenko, M. Niethammer, J. S. Marron, D. Borland, J. T. Woosley, X. Guan, C. Schmitt, and N. E. Thomas, "A method for normalizing histology slides for quantitative analysis," *Proceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 1107–1110, Jun. 2009.

[16] Wikipedia contributors. (2021, Apr.) H&E stain. Wikipedia. [Online]. Available: https://en.wikipedia.org/w/index.php?title=H%26E_stain&oldid=1017091635

[17] (2021, Apr.) What is H&E? University of Leeds Faculty of Biological Sciences, Leeds, United Kingdom. [Online]. Available: https://histology.leeds.ac.uk/what-is-histology/H_and_E.php

[18] M. R. Wick, "The hematoxylin and eosin stain in anatomic pathology—An often-neglected focus of quality assurance in the laboratory," *Seminars in Diagnostic Pathology*, vol. 36, no. 5, pp. 303–311, Sep. 2019.

[19] A. Nanci, "Chapter 1 - structure of the oral tissues," in *Ten Cate's Oral Histology*, 8th ed. St. Louis, Missouri: Mosby, 2013, pp. 1–13.

[20] ——, "Chapter 12 - oral mucosa," in *Ten Cate's Oral Histology*, 8th ed. St. Louis, Missouri: Mosby, 2013, pp. 278–310.

[21] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *Proceedings of the IEEE Computer Graphics and Applications*, vol. 21, no. 5, pp. 34–41, Oct. 2001.

[22] M. T. Shaban, C. Baur, N. Navab, and S. Albarqouni, "Staingan: Stain style transfer for digital histological images," *IEEE 16th International Symposium on Biomedical Imaging*, pp. 953–956, Apr. 2019.

[23] D. Magee. (2021, Apr.) Stain normalization toolbox. Department of Computer Science at the University of Warwick, Warwick, United Kingdom. [Online]. Available: https://warwick.ac.uk/fac/cross_fac/tia/software/sntoolbox

[24] A. M. Khan, N. Rajpoot, D. Treanor, and D. Magee, "A nonlinear mapping approach to stain normalization in digital histopathology images using image-specific color deconvolution," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 6, pp. 1729–1738, Jan. 2014.

[25] S. J. Russell and P. Norvig, *Artificial Intelligence A Modern Approach*. Englewood Cliff, New Jersey: Prentice Hall, 1995.

[26] S. T. H. Kieu, A. Bade, M. H. A. Hijazi, and H. Kolivand, "A Survey of Deep Learning for Lung Disease Detection on Medical Images: State-of-the-Art, Taxonomy, Issues and Future Directions," *Journal of Imaging*, vol. 6, no. 12, Dec. 2020.

[27] A. S. Sultan, M. A. Elgharib, T. Tavares, M. Jessri, and J. R. Basile, "The use of artificial intelligence, machine learning and deep learning in oncologic histopathology," *Journal of Oral Pathology and Medicine*, vol. 49, no. 9, pp. 849–856, May 2020.

[28] M. K. K. Niazi, A. V. Parwani, and M. N. Gurcan, "Digital pathology and artificial intelligence," *The Lancet Oncology*, vol. 20, no. 5, pp. e253–e261, May 2019.

[29] M. Borovcnik, H.-J. Bentz, and R. Kapadia, "A Probabilistic Perspective," in *Chance Encounters: Probability in Education*. Dordrecht, Netherlands: Springer-Verlag, 1991, pp. 27–71.

[30] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, Massachusetts: MIT Press, 2016.

[31] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, Mar. 1989.

[32] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," *International Conference on Machine Learning*, pp. 6105–6114, Jun. 2019.

[33] J. Miao and W. Zhu, "Precision-recall curve (PRC) classification trees," *Evolutionary Intelligence*, pp. 1–25, Apr. 2021.

[34] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information processing & management*, vol. 45, no. 4, pp. 427–437, Jul. 2009.

[35] A. Hinterreiter, P. Ruch, H. Stitz, M. Ennemoser, J. Bernard, H. Strobelt, and M. Streit, "ConfusionFlow: A model-agnostic visualization for temporal analysis of classifier confusion," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–15, Jul. 2020.

[36] S. Raschka, "Model evaluation, model selection, and algorithm selection in machine learning," *arXiv preprint arXiv:1811.12808*, Nov. 2018.

[37] E. Stevens, L. Antiga, and T. Viehmann, "Chapter 1," in *Deep learning with PyTorch*, 1st ed. Shelter Island, New York: Manning Publications Company, 2020.

[38] R. S. Michalski J. G. Carbonell T. M. Mitchell, *Machine Learning: An Artificial Intelligence Approach*. Berlin, Germany: Springer Verlag, 1983.

[39] E. C. Too, L. Yujian, P. K. Gadosey, S. Njuki, and F. Essaf, "Performance analysis of nonlinear activation function in convolution neural network for image classification," *International Journal of Computational Science and Engineering*, vol. 21, no. 4, pp. 522–535, Apr. 2020.

[40] Y. Lecun, "1.1 Deep Learning Hardware: Past, Present, and Future," *IEEE International Solid-State Circuits Conference*, pp. 12–19, Feb. 2019.

[41] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[42] N. S. Keskar, J. Nocedal, P. T. P. Tang, D. Mudigere, and M. Smelyanskiy, "On large-batch training for deep learning: Generalization gap and sharp minima," *ICLR 2017*, pp. 1–16, Feb. 2017.

[43] F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte, "Breast cancer histopathological image classification using Convolutional Neural Networks," *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 2560–2567, Jul. 2016.

[44] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, vol. 6, no. 1, Jul. 2019.

[45] A. Maier, C. Syben, T. Lasser, and C. Riess, "A gentle introduction to deep learning in medical image processing," *Zeitschrift fur Medizinische Physik*, vol. 29, no. 2, pp. 86–101, May 2019.

[46] K. O'shea and R. Nash, "An Introduction to Convolutional Neural Networks," *arXiv preprint arXiv:1511.08458v*, Nov. 2015.

[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, Jan. 2012.

[48] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Proceedings of the The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, Jan. 2018.

[49] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 248–255, Jun. 2009.

[50] P. Hensman and D. Masko, *The Impact of Imbalanced Training Data for Convolutional Neural Networks*, Bachelor's thesis, KTH, Stockholm, Sweden, May 2015.

[51] C. Xie, M. Tan, B. Gong, J. Wang, A. L. Yuille, and Q. V. Le, "Adversarial examples improve image recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 816–825, Jun 2020.

[52] D. Olson and D. Delen, *Advanced Data Mining Techniques*. Heidelberg, Germany: Springer-Verlag, 2008.

[53] D. H. Wolpert and W. G. Macready, "No Free Lunch Theorems for Optimization," *Natural Computing Series*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[54] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Computation*, vol. 10, no. 7, pp. 1895–1923, Oct. 1998.

[55] M. Lee, J. Jeon, and H. Lee, "Explainable AI for domain experts: a post Hoc analysis of deep learning for defect classification of TFT–LCD panels," *Journal of Intelligent Manufacturing*, Mar. 2021.

[56] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, pp. 82–115, Jun. 2020.

[57] N. Kumar, R. Verma, S. Sharma, S. Bhargava, A. Vahadane, and A. Sethi, "A dataset and a technique for generalized nuclear segmentation for computational pathology," *IEEE Transactions on Medical Imaging*, vol. 36, no. 7, pp. 1550–1560, Mar. 2017.

[58] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Oct. 2015.

[59] P. Naylor, M. Laé, F. Reyal, and T. Walter, "Nuclei segmentation in histopathology images using deep neural networks," *IEEE 14th International Symposium on Biomedical Imaging*, pp. 933–936, Apr. 2017.

[60] V. Chouhan, S. K. Singh, A. Khamparia, D. Gupta, P. Tiwari, C. Moreira, R. Damaševičius, and V. H. C. de Albuquerque, "A novel transfer learning based approach for pneumonia detection in chest X-ray images," *Applied Sciences*, vol. 10, no. 2, Jan. 2020.

[61] M. Shorfuzzaman and M. Masud, "On the detection of covid-19 from chest x-ray images using cnn-based transfer learning," *Computers, Materials and Continua*, vol. 64, no. 3, pp. 1359–1381, Jun. 2020.

[62] P. Lakhani and B. Sundaram, "THORACIC IMAGING: Deep Learning at Chest Radiography Lakhani and Sundaram," *Radiology*, vol. 284, no. 2, pp. 574–582, Aug. 2017.

[63] R. Hooda, A. Mittal, and S. Sofat, "Automated TB classification using ensemble of deep architectures," *Multimedia Tools and Applications*, vol. 78, no. 22, pp. 31 515–31 532, Nov. 2019.

[64] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," *arXiv preprint arXiv:1503.02531*, Mar. 2015.

[65] M. T. Islam, M. A. Aowal, A. T. Minhaz, and K. Ashraf, "Abnormality detection and localization in chest x-rays using deep convolutional neural networks," *arXiv preprint arXiv:1705.09850*, May 2017.

[66] T. Lin. (2018, Apr.) LabelIMG. GitHub. [Online]. Available: https://github.com/tzutalin/labelImg

[67] Wikipedia contributors. (2021, May) Bicubic interpolation. Wikipedia. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Bicubic_interpolation&oldid=1005441439

[68] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 6, pp. 1153–1160, Dec. 1981.

[69] L. Melas-Kyriazi. (2020, Apr.) Efficientnet-pytorch. GitHub. [Online]. Available: https://github.com/lukemelas/EfficientNet-PyTorch

[70] (2021, Apr.) PyTorch. Facebook. [Online]. Available: https://ai.facebook.com/tools/pytorch

[71] (2021, Apr.) Alvis, Swedish National Infrastructure for Computing (SNIC). Chalmers University of Technology, Gothenburg, Sweden. [Online]. Available: https://www.snic.se/resources/compute-resources/alvis/

[72] A. Aillet and F. Frisk. (2021) Bachelor thesis documentation: Assessing the impact of stain normalization on a cell classification model in digital histopathology. Github. [Online]. Available: https://github.com/filipfusk/BscThesis

[73] D. Komura and S. Ishikawa, "Machine Learning Methods for Histopathological Image Analysis," *Computational and Structural Biotechnology Journal*, vol. 16, pp. 34–42, Jan. 2018.

CONTEXT R
EMBEDDED SYSTEMS

POPULAR DESCRIPTION

Embedded systems: Putting your worries to bed

The modern transistor was invented in 1947, and has since then decreased in size exponentially every year. Nowadays transistors are so small that you could fit 100.000 transistors on the cross section of a regular human hair. This development has allowed embedded devices to become more powerful, compact and readily available than ever before.

Without embedded systems we would not have smart devices such as, cell phones, TVs, wireless headsets or calculators. Basically our modern society would not be the same without them. As the size of computers have decreased they have become embedded in more and more devices. Even simple appliances like coffee machines and fridges are becoming smart by including processors that enable more advanced functionalities and connectivity. Embedded devices are not necessarily smart, but a smart device necessarily needs an embedded system.

Lately the field of Internet of things (IoT) has taken the world by storm. IoT enabled devices help make our homes smarter, and our lives easier. Smart fridges can make everyday burdens like shopping for groceries easier, and intelligent home security systems can increase the safety of our homes. Smart watches and other wearable devices have grown in popularity immensely in the last years due to the performance they can nowadays provide. Modern smart watches have the ability to measure your health, assist you on your workouts and analyze your sleeping.

Our devices are becoming smarter at a rate so rapid that it is near impossible to tell exactly how far embedded systems will take our devices in the future. One thing is for sure, though: they are here to stay and will definitely continue to have a huge impact on our lives in the coming years.

SUMMARY OF PROJECT RESULTS

In modern society, embedded systems perform an increasing amount of essential functions in areas like communication, transportation and the Internet of things. Almost every piece of technology we use on a daily basis contains an embedded system. The rapidly increasing demand for smaller and more power efficient computing is making the development process more complex. The interconnection of systems within local and wide-area networks results in new possibilities, albeit in combination with new vulnerabilities.

The objective of project group R1 was to use correlation analysis, combined with previously investigated methods with deep learning, to achieve a more automated side-channel attack (SCA). This type of attack is based on information leakage caused by the hardware implementation and not the encryption algorithm itself. Examples of information channels used for SCAs include power consumption and electromagnetic radiation. For this project, the target was a certain implementation of the advanced encryption standard (AES) algorithm. The AES encryption algorithm is one of the most used in technology and communication today.

The AES algorithm was running on a development board with a bluetooth radio and the SCA was utilizing the electromagnetic radiation. After the leaking radiation was sampled into a digital signal, a correlation method was used to identify the signal segments that correspond to different steps of the AES algorithm.

Data from the project's testing phase are showing promising results for the correlation method, especially when the level of background noise is low. The results are not as good when background noise level is significantly higher than the targeted side channel signal. This might be a weakness of the correlation method, which gives reason to investigate other encryption identification methods in the future, for example using deep learning.

Project group R2 had the aim to create a middleware that enables embedded system developers to use multiple connected devices as a single device. The middleware acts as a software interface that removes the need to manually pack and send data between devices. The software used a controlled area network (CAN) as its underlying means of communication, as this is the primary protocol used in the automotive industry. This middleware works by using a model to generate code for each of the devices on the network. Software generated by the middleware is then responsible for optimizing bandwidth and scheduling signals throughout the network.

Primarily, the system was mainly focusing on FreeRTOS platforms, but has potential to be extended to multiple platforms, i.e. LitmusRT based systems. Future users of this middleware may add additional communication protocols, as the system is designed with this in mind. However, schedulability may only be guaranteed for CAN.

The project group in R4 has designed and tested a telemetry unit for an electric driverless vehicle with the purpose of simplifying the testing process of new systems in such vehicles. The focus has been on the design and verification of a sub-1 GHz low-power radio link from a printed circuit board level, as well as the integration between hardware and embedded software to run such a system. Additionally, the project group implemented integration between the collection of sensory data on the vehicles data bus (CAN) and a Radio-Frequency link, as well as presenting said data to the end-user in a user friendly way.

A first revision prototype has given promising results for a proof of concept. As a whole, all subsystems work individually as well as together, although further optimization will be required to achieve the desired range and throughput. In future projects different methods of achieving higher data rates could be investigated, making it possible to send more bandwidth-heavy information such as video or audio from the vehicle in real-time. Different frequency bands and modulation- or error correcting algorithms could also be analyzed further to make the communication link more robust to disturbance. For example intelligent channel sensing and frequency hopping algorithms could be investigated.

IMPACT ON SOCIETY AND ENVIRONMENT

The main environmental impact of embedded systems comes from manufacturing and transportation, as the embedded system often consumes negligible amounts of power during their lifetime. The manufacturing of said systems often uses rare-earth minerals and other non-renewable resources, which has a significant environmental impact. In some cases regarding the mining of rare-earth minerals it also introduces a moral dilemma, as the cheap minerals are mined in conditions that are not up to humane standards. Furthermore, the manufacturing supply chain is very long and complicated, making it near impossible to know the exact environmental and societal implications of manufacturing an embedded device.

Both the production phase and the supply chain for raw materials often have a big ecological impact on animals and nature. The working conditions at the production facilities may also affect the health of workers negatively. Emissions of toxic substances therefore need to be monitored and likewise the working conditions for the workers. Today a lot of the production is taking place far away from the end user, which gives even lower transparency and awareness of the social and ecological impact that every product has. To address this issue, consumers have to clearly demand sustainable products. However, the designing engineers, manufacturers and authorities also have a big responsibility to make sustainable choices as well as to facilitate the customer's insight in the production process.

As for recycling of embedded devices, while possible to a certain degree it is often not economically justified. The silicon that make up these chips is comparatively cheap to produce and difficult to recycle. The recycling of copper, gold and other metals within them on the other hand is more lucrative and can compensate for the environmental impact of manufacturing to some degree.

In the industry, embedded systems contribute to many advantages, both economical and technological. They add more advanced sensors and communication systems, which are necessary for autonomous vehicles and the internet of things (IoT). This can greatly impact the efficiency for testing these systems and greatly reduce the risk of failures. It can also result in large savings in terms of resources and money. Furthermore, the extraction of more data from similar systems could speed up development of new systems.

From a societal point of view, embedded devices at large have a positive impact on our daily lives. IoT devices can make our lives more comfortable by providing us with new quality-of-life services. Within medical applications, they can increase the level of care and make it more readily available by adding the capability of autonomy and data collection to the process, although this introduces the concern of privacy. The projects in this context are aimed at furthering the development of new embedded systems.

In R1 we want to show that if devices are not carefully designed they might unintentionally leak sensitive information in side-channels, for example in their power consumption or in electromagnetic radiation. Even though this might not be an imminent threat to the individual, this sheds light on a broader problem. An embedded device used to perform a trivial task might not get the security attention needed due to time and cost constraints. For example, the device can later serve as an attack entrypoint to a network. Taking side-channels into account may however lead to an increased cost for both the manufacturer and the consumer, which can't be justified for all implementations.

In R2 we create means for easier development of distributed control systems in cars. The provided abstraction level may reduce the developing time of complex interconnected systems. By making the development process easier and less error prone, better products can come to market faster. Furthermore, the distribution of computing power closer to the data source makes fault isolation and maintenance easier. With a distributed system, individual nodes can work independently and enable individual parts of a faulty system to be changed or upgraded, rather than the whole system. This leads to reduced waste and less resources being used.

In R4 we investigate methods of wireless testing, which would allow for easier runtime model testing and verification. This would lead to more sophisticated models within driverless vehicle technology by speeding up and simplifying development and testing time. Anomalies in the model, or faults in hardware of real vehicles could be detected in runtime and be compared with expected behavior of a digital model to prevent accidents, reduce maintenance costs and further optimize models.

As a whole, the projects have contributed to the development of modern, secure and power efficient embedded systems. All groups in this context have aimed to investigate embedded systems and how they could be designed to adapt and meet the demands of a complex, continually evolving environment.

Using Correlation Analysis to Locate Encryption Activity in Electromagnetic Side-Channels

Simon Weideskog and Tore Johansson

***Abstract*—Physical implementations of cryptographic algorithms can leak sensitive information through different kinds of side-channels. This information can potentially be used for recovering the secret key used in the algorithms. Recently, side-channel attacks on CPU implementations of Advanced Encryption Standard (AES) has been presented. Some of these attacks use far-field electromagnetic radiation.**

In this thesis, we investigate if cross correlation can be used to locate the part of the signal corresponding to the execution of the AES-encryption. We gather side-channel signal data containing encryption activity to create and test multiple templates for correlation. By evaluating the performance of the templates in different scenarios, we conclude that the method is useful and relatively easy to implement compared to previous methods. Furthermore, our extraction method has a low execution time which gives it potential to be used in real-time attacks.

***Sammanfattning*—Fysiska implementationer av krypteringsalgoritmer kan läcka känslig information genom olika typer av sidokanaler. Informationen kan potentiellt användas för att återskapa algoritmernas hemliga krypteringsnycklar. Senaste tiden har sidokanalsattacker mot CPU-implementationer av Advanced Encryption Standard (AES) presenterats. Vissa av dessa attacker nyttjar utstrålande elektromagnetiska fjärrfält.**

I denna rapport undersöker vi om korskorrelation kan användas för att hitta tidpunkten i signalen då AES-krypteringen genomfördes. Vi samlar in signaldata från sidokanalen som innehåller krypteringsaktivitet, detta för att skapa och testa olika korrelationsmallar. Genom att utvärdera mallarnas prestanda i olika scenarion kan vi sluta oss till att denna metod är användbar samt lätt att implementera jämfört med tidigare metoder. Därtill visar vår metods korta körtid att den har potential att användas i en realtidsattack.

***Index Terms*—AES, Cross correlation, Side-channel, SCA, Encryption, EM, Far-field**

Supervisors: Elena Dubrova

TRITA number: TRITA-EECS-EX-2021:199

I. Introduction

Security in today's society is a very complex subject. When talking about device security and privacy, software has the main focus in mass media with frequently used terms like 'cyber attacks' and 'hacking'. Hardware on the other hand is often overlooked, even if hardware related flaws also can serve as potential attack points [1]. Side-channel attacks (SCA) are one of the prominent threats in this area. These attacks are often targeted at cryptographic algorithms and by observing activity on side-channels, an attacker can gain sensitive information from the device [2]. Encryption keys are usually the information of interest.

The demand for small and cheap devices leads to side-channels being more exposed to a potential attack. If the device under attack incorporates a wireless radio transmitter, the side-channel information can also be amplified and propagate in far-field electromagnetic (EM) radiation. This enables an attacker to operate from a distance, leaving the target unaware of the ongoing attack.

In [3] the authors show that EM far-fields from a Bluetooth radio incorporated in the same dye as a CPU-core can leak information about the internal activity. They also show that this information can be used in an attack against the running encryption algorithm. To locate the sensitive information the authors manually identified a frequency component in the signal that precedes each encryption. From that component they extract what they call a trigger. However, the trigger only gives an estimate of where the encryption starts and to further align the traces they use correlation. After this, they use template and correlation attacks to recover the key.

Later on, the authors of [4] show that it is possible to perform a ciphertext only attack using deep learning (DL) instead of correlation. They manage to recover the secret key with information from less than 500 encryptions, captured from side-channels in the EM spectrum.

Our contribution to the field is to show that cross correlation can be used to extract encryption activity from a side-channel signal. With our method the needed information for an SCA can be extracted in a single step. This gives a simpler alternative to the trigger method used in previous work. We also provide means of using the proposed method in a more complex environment. In the discussion we give a suggestion on how a template should be created for using the method in future attacks.

In section II we give a summary of the key concepts that are involved in our experiments. After that background we describe in section III how the traces are collected, followed by an explanation of how we create the templates for cross correlation in section IV. In section V we describe our method for extracting encryption blocks from a trace. We explain our experiments and show our results in section VI, then discuss them in section VII.

II. Background

A. Terminology

Trace: Time discrete sampled signal of an electromagnetic side-channel.

AES-128: Advanced Encryption Standard using 128-bit key. More information under section II-B.

Plaintext: Non-encrypted text.
Ciphertext: Encrypted text, in this case using AES-128 algorithm.
Encryption block: Part of a trace where encryption activity from an entire encryption appears.

B. AES-128

The Advanced Encryption Standard (AES) is a cryptographic algorithm standardized by National Institute of Standards and Technology in 2001 [5]. AES is a symmetric block cipher that takes a 128-bit plaintext message and encrypting it with a key of length n ($n \in \{128, 192, 256\}$), creating a 128-bit ciphertext. AES-128 is computed in 10 rounds where each round, but the last, performs byte substitution, shift row, mix column and adds a round key. The round key is different for each round and derived from the original key. The last round omits the mix columns operation. We are using AES in electronic codebook mode, in which the message is divided into blocks and each block is encrypted separately.

C. Side-channel attack

Side-channel attacks were introduced in the late 90s by Paul Kocher. He showed that timing of a cryptographic implementation could be leaking information about the secret key [6]. Kocher also introduced power analysis, which lays a foundation for electromagnetic (far-field) attacks [7]. The goal of an SCA is often to recover the secret key used in cryptographic algorithms by analyzing information from a side-channel (e.g. power consumption, timing characteristics or EM emissions). To recover a key $\mathcal{K}$, the attacker would use a known input $\mathcal{P}$ (plaintext or ciphertext) and a set of corresponding physical measurements $\mathcal{T}$ from the chosen side-channel. By using statistical methods and analyzing the connection between $\mathcal{P}$ and $\mathcal{T}$, the attacker would end up with a probability for all possible keys. Using the right estimation metrics, the correct key can be selected [8].

D. EM emissions as a side-channel

Leaking far-field EM emissions from a circuit combining digital and analog circuitry is caused by capacitive substrate coupling. As described in [3], if the isolation between the digital and analog part of the circuit is insufficient, the activity in the cryptoblock will be coupled with the radio frequency (RF) block. The CPU clock generates a square wave noise which is modulated by the cryptographic computations. This modulated signal will, due to the poor isolation, leak to the analog part of the circuit. In the analog part the signal will be further modulated and amplified by the RF block, before unintentionally being transmitted via the antenna. The main blocks involved in this leakage are included in the block diagram of Fig. 1.

By analyzing the resulting signal in the frequency domain, we can understand how information about the cryptographic computations can be obtained from the emitted EM field. The details are explained in [8]. If the receiver is set to receive at $f_r = nf_{clock} + f_{bluetooth}$, the side-channel from the cryptoblock can be retrieved by low-pass filtering the signal.

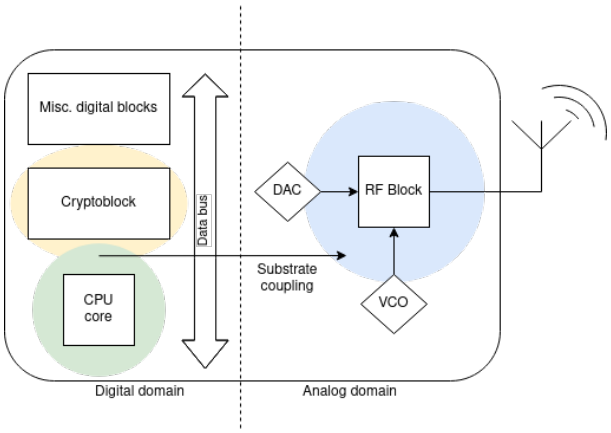


Fig. 1. High level illustration of the blocks that cause EM side-channel emission in mixed signal circuits.

TABLE I
EQUIPMENT USED TO CAPTURE TRACES.

For transmission	- Nordic nRF52 DK board
	- nRF52832
For receiving	- Ettus Research USRP N210 SDR
	- Coaxial cable
	- 2.4 GHz monopole antenna

III. TRACE ACQUISITION

In the following sections we describe how we capture the signal from a target device.

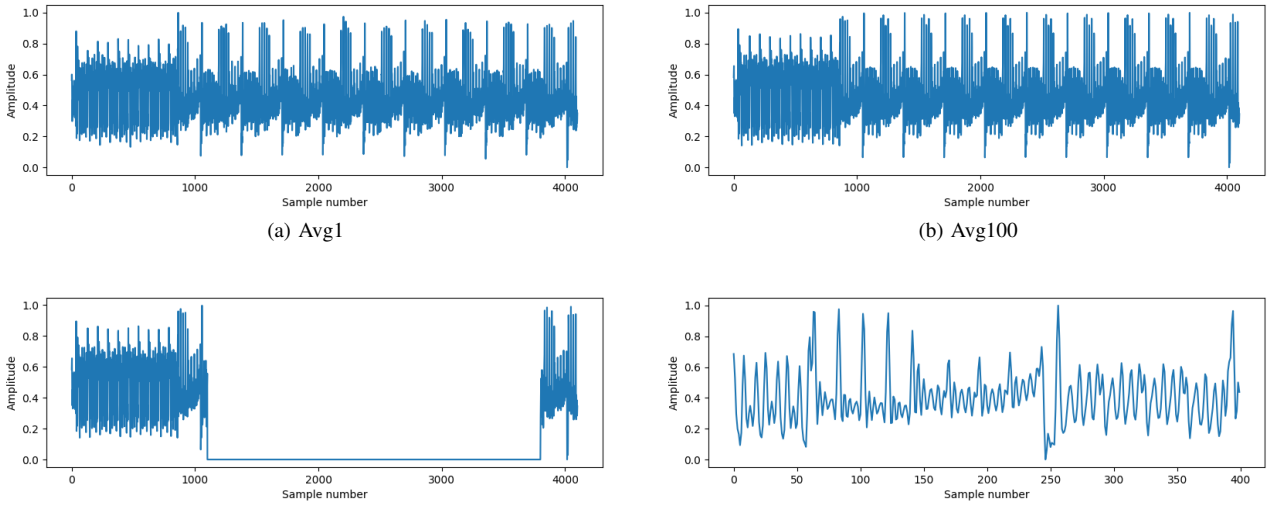
A. Measurement setup

We use the equipment listed in Tab. I to capture all traces. The Nordic nRF52 DK board is used as the target device. The board has an nRF52832 chip mounted which contains an ARM Cortex M4 CPU running at $f_{clock} = 64\,\text{MHz}$, together with a Bluetooth radio. The nRF52832 is running an implementation of *TinyAES* from [9] with an 128-bit key. The device is set to continuously run the Bluetooth radio at $f_{bluetooth} = 2.4\,\text{GHz}$. The Ettus Research USRP N210 is a software defined radio (SDR) capable of receiving and transmitting on frequencies up to 6 GHz. Its receiving center frequency is set to $2f_{clock} + f_{bluetooth} = 2.528\,\text{GHz}$ and the sample rate to $f_s = 5\,\text{MHz}$. We connect the target device and SDR via the coaxial cable for most of the capturing and in one case we use the antenna.

B. Acquisition procedure

We set up the target device to run its Bluetooth radio continuously and to periodically perform encryption with a fixed key and random plaintext. The signal received by the SDR is amplitude demodulated, then stored on a computer. The acquired trace contains the leaked side-channel information and in the time domain, plotted in Fig. 2a, one can see the periodical encryption activity. We capture three sets of data, which are all presented below.

1) For template: This set contains 100 000 traces using 50 keys and 2000 random plaintexts. Each trace contains one encryption.



(a) Avg1

(b) Avg100

(c) Avg100 first & last round

(d) Avg100 first round

Fig. 2. Some of the tested templates, normalized using min-max normalization.

2) For testing: This set contains $5 \cdot 10\ 000$ traces using five keys and random plaintexts. In each trace the encryption is repeated ten times with the same key and plaintext, resulting in a total of $500\ 000$ encryptions.

3) Noisy test traces: This set contains traces that are captured with both cable and antenna. When capturing these traces, the target device was instructed to perform more tasks than just the encryption. This results in other activities appearing in the traces.

IV. Template for correlation

To identify the encryption blocks in the traces a template to correlate with is needed. The template must include characteristics that are mutual for all encryption blocks. It also must be specific enough that the cross correlation between the template and the traces reaches its maximum value if and only if the template completely overlaps an encryption block.

Initially we manually cut one encryption block from a trace as the first template, called *Avg1* (average of one). Using the correlation method described in section V together with the template *Avg1*, we extract more encryption blocks from the data set *For template*. With the larger set of encryption blocks we create different types of templates.

A. Averaging

We use three different levels of averaging for the templates:

- *Avg1* – average of 1 encryption (in practice no averaging);
- *Avg100* – average of 100 encryptions (5 different keys and 20 different plaintexts);
- *Avg100K* – average of 100 000 encryptions (50 different keys and 2000 different plaintexts).

B. Different parts

Our templates include and exclude different parts of the encryption blocks. The different combinations we use are:

- entire encryption blocks (Fig. 2a and 2b);
- first and last encryption rounds with zeros between (Fig. 2c);
- only the first encryption rounds (Fig. 2d);
- only the last encryption rounds.

C. Template normalization

In total we have 12 templates to test. They are all listed in Tab. II and four of them are illustrated in Fig. 2. To be able to compare all templates and their performance we normalize them using min-max scaling. Given a template $\mathbf{\Gamma} = (\gamma_1, \gamma_2, ..., \gamma_n)$, the normalized template $\mathbf{\Gamma'} = (\gamma'_1, \gamma'_2, ..., \gamma'_n)$ is given as

$$\gamma'_i = \frac{\gamma_i - min(\mathbf{\Gamma})}{max(\mathbf{\Gamma}) - min(\mathbf{\Gamma})},\ i \in \{1, 2, ..., n\}. \tag{1}$$

V. Extracting encryption blocks

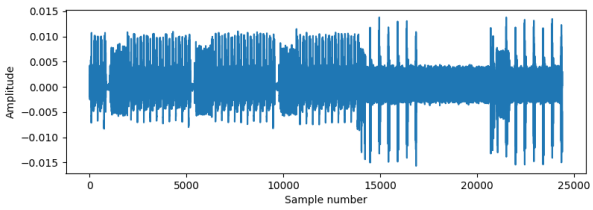
In the following sections we describe how we extract the encryption blocks from captured traces. The extraction is done in several steps to maximize accuracy and performance.

A. Cross correlation and envelope array

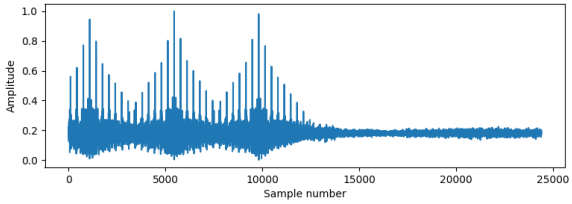
To locate the encryption blocks in a trace $\boldsymbol{\mathcal{T}} = (\tau_1, \tau_2, ..., \tau_m)$, the cross correlation with a template $\mathbf{\Gamma} = (\gamma_1, \gamma_2, ..., \gamma_n)$ is calculated. Since all values in $\boldsymbol{\mathcal{T}}$ and $\mathbf{\Gamma}$ are real numbers, each element ρ_k of the cross correlation can be calculated as

$$\rho_k = \sum_{l=1}^{m} \tau_l \gamma_{l-k+n} \tag{2}$$

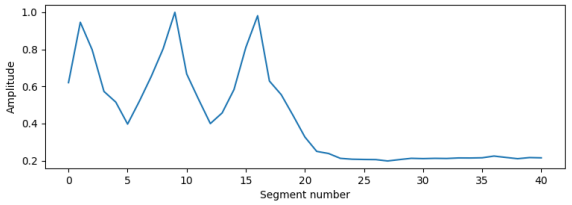
where $\gamma_i = 0$ for $i \notin \{1, 2, ..., n\}$. The resulting array $\boldsymbol{r} = (\rho_1, \rho_2, ..., \rho_{m+n})$ contains the correlation values when the template is shifted to sample k. For visualization we plot one part of a trace in Fig. 3a, with its corresponding correlation array in Fig. 3b.



(a) Part of a trace



(b) Correlation array



(c) Envelope array

Fig. 3. Part of a trace from the data set *Noisy test traces* with three visible encryption blocks. Also plotted is the corresponding correlation (b) and envelope (c) arrays. The template used for this figure is *Avg100*.

Since the high correlation values are the ones of interest, we have to find the local maximums. Due to the correlation array containing an arbitrary number of local maximums, the correlation array is converted to an envelope array. An example is shown in Fig. 3c. We create the envelope by splitting the correlation array into $\frac{m}{l}$ number of segments, where l is the length of each segment and m is the size of the correlation array. The envelope array is then populated by taking the maximum value and its index from each of these segments.

B. Using the envelope array

To extract the encryption blocks from a trace, we need the indices where the maximum correlation occurs. Since the trace can contain several encryption blocks, we need to consider the local maximums in the envelope array. To find the ones that are relevant we define a trigger level. Local maximums above the trigger level are considered as identified encryption blocks while local maximums below the level are ignored. We let the trigger level be the standard deviation of the correlation array, multiplied with a scalar. Since the relation between correlation peak value and standard deviation (Fig. 4) has a consistent order of magnitude, we can use a similar multiplier for all combinations of templates and traces.

As stated in section V-A, relevant indices are stored in the envelope array. Therefore, the final step before extracting the identified encryption blocks is to retrieve their corresponding starting indices from the envelope. When we have these

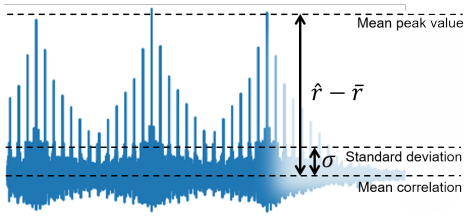


Fig. 4. Illustration of the metrics used to calculate peak significance.

indices, we extract the number of samples that correspond to the known length of an encryption block and save them into a new trace. Traces of specific rounds can also be extracted in the same way, based on known starting points of the rounds within the encryption blocks.

VI. Experimental results

A. Correlation with different templates

For testing the 12 templates we use the data set *For testing*. We test all templates using the extraction method described in section V, with an envelope segment size $l = 600$. The results can be seen in Tab. II, where accuracy $\mathcal{A}$ is calculated as the percentage

$$\mathcal{A} = 100 \cdot \frac{N_{found}}{N_{total}}. \tag{3}$$

N_{found} is the number of encryption blocks found and $N_{total} = 499\ 889$, the total number of encryptions in the data set. Peak significance $\mathcal{S}$ is calculated as

$$\mathcal{S} = \frac{\hat{r} - \bar{r}}{\sigma} \tag{4}$$

where $\hat{r}$ is the mean peak value and $\bar{r}$ is the mean correlation. In Fig. 4 these values are illustrated.

B. Adding noise

To test the robustness of our method we introduce additive white Gaussian noise to the data set *For testing*. The white noise has mean $\mu = 0$ and standard deviation $\sigma = 0.003$. This standard deviation is approximately the same as the standard deviation of the traces in *For testing*. With all the templates in Tab. II that has an accuracy $\mathcal{A} = 100\%$ we run the same test as in section VI-A. The results can be seen in Tab. III.

C. Antenna trace

To test the method in a more challenging environment we use our third data set, *Noisy test traces*. In the traces captured with antenna there is disturbances, as can be seen in Fig. 5a. The signal strength of these disturbances is up to 40 times stronger than encryption blocks in the same trace. Because of the high signal strengths, these disturbances result in high correlation values despite not being signals of interest. To address this issue we introduce a normalization of the envelope array, compensating for the varying signal strength in the trace. For an envelope array with segment length l, the normalization

TABLE II
STATISTICS FOR TEMPLATES, TESTED WITH TRACES CAPTURED THROUGH CABLE.

Template name	Accuracy ($\mathcal{A}$)	Mean correlation	Mean peak value	Standard deviation	Peak significance ($\mathcal{S}$)
Avg1	100	1.3346	3.0333	0.074	22.96
Avg1 first & last	100	0.4553	1.1634	0.032	22.13
Avg1 first round	72.97	0.1154	0.2867	0.014	12.24
Avg1 last round	100.02	0.1375	0.3058	0.011	15.3
Avg100	100	1.3672	3.1796	0.079	22.94
Avg100 first & last	100	0.4663	1.2207	0.035	21.55
Avg100 first round	61.57	0.1212	0.3035	0.015	12.15
Avg100 last round	100.02	0.1347	0.3055	0.012	14.23
Avg100k	100	1.3701	3.1867	0.079	22.99
Avg100k first & last	100	0.4673	1.2234	0.035	21.6
Avg100k first round	59.94	0.1213	0.3031	0.015	12.12
Avg100k last round	100.02	0.1347	0.3057	0.012	14.25

is based on a larger segment $\boldsymbol{B}_k \in \mathbb{R}^L$ of the trace $\boldsymbol{\mathcal{T}} \in \mathbb{R}^m$. The elements of $\boldsymbol{B}_k$ are defined as

$$B_{k,i} = |\boldsymbol{\mathcal{T}}_{k-L/2+l+i}|, \ i \in \{1, 2, ..., L\}. \tag{5}$$

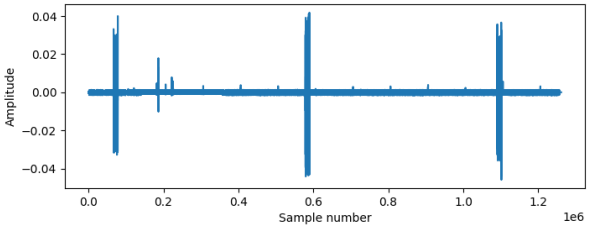
The normalization coefficient c_k is calculated as

$$c_k = \frac{1}{100 \max(\boldsymbol{B}_k)} \tag{6}$$

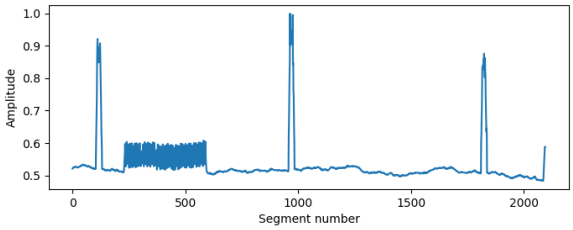
and the final elements e_j stored in the envelope array is

$$e_j = (\rho_k)^5 c_k, \ (k = jl, \ j \in \{1, 2, ..., \frac{m}{l}\}). \tag{7}$$

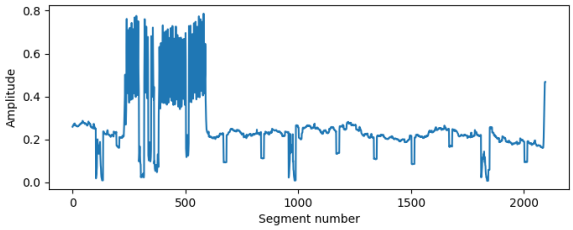
With this normalization the parts of the envelope with low correlation value in relation to signal strength is suppressed, which can be seen in Fig. 5. By extracting encryption blocks using this normalized envelope we get an accuracy $\mathcal{A} = 88.4\%$, for the traces captured with antenna. In this case we used a normalization segment size $L = 8000$. The extraction process takes approximately $10\,\mathrm{s}$ for a data set containing $12.6\,\mathrm{s}$ of recorded data.



(a) Antenna trace



(b) Envelope array



(c) Normalized envelope array

Fig. 5. Antenna trace (a) from the data set *Noisy test traces* with its corresponding envelope array, both not normalized (b) and normalized (c).

VII. DISCUSSION

In this section we will discuss the results from the previous section. We reason about which template is best and why. We also bring up potential flaws in our testing and the data that we capture.

A. The results

By comparing the results presented in Tab. II and III we can deduce which template performs the best. We see from Tab. II that three templates have an accuracy $\mathcal{A} < 75\%$, hence the reason they are not included in Tab. III. After introducing noise in the data set and running the tests again, the performance of the templates on noisy signals can be evaluated. Here we see that templates only including the last round find less than 50% of all the encryptions, regardless of template averaging. This is reflected in the peak significance which is relatively low, indicating that the peaks are not as distinct.

In the second experiment, six of the templates still had an accuracy of approximately 100%. We argue that the template among these six that also has the highest peak significance would perform best for an attack. When also considering the number of traces needed for each template, our suggestion for future adopters is to use a template similar to *Avg100* (Fig. 2b).

The trigger level is directly impacting the measured value of the accuracy. When the trace is noisy, the choice of trigger

TABLE III
STATISTICS FOR TEMPLATES, TESTED WITH TRACES CAPTURED THROUGH CABLE WITH ADDED NOISE.

Template name	Accuracy ($\mathcal{A}$)	Mean correlation	Mean peak value	Standard deviation	Peak significance ($\mathcal{S}$)
Avg1	100	0.225	0.936	0.048	14.94
Avg1 first & last	99.99	0.301	0.9258	0.054	11.63
Avg1 last round	5.3	0.382	0.9961	0.089	6.86
Avg100	100	0.22	0.9377	0.047	15.26
Avg100 first & last	100	0.302	0.9281	0.053	11.91
Avg100 last round	42.68	0.379	0.9555	0.088	6.54
Avg100k	99.71	0.2203	0.9381	0.047	15.24
Avg100k first & last	100	0.3012	0.9279	0.053	11.91
Avg100k last round	42.74	0.3789	0.9555	0.088	6.54

multiplier has an even higher impact since the amplitude of the correlation peaks vary more in relation to each other and are not as distinct. In Tab. III we used different trigger multipliers for *Avg1 last round*, *Avg100 last round* and *Avg100K last round*, 6 for the first and 7 for the two later templates. By lowering the trigger level with a factor of $\frac{6}{7}$, we got an approximate increase in number of found traces by a factor 8.

It is worth noting that we have not been able to verify whether we get false positives in our results. This means that the presented accuracy might be higher than the actual and explains some of the results in Tab. II. This is mainly a problem when the peak significance is low or the envelope is noisy and the trigger level is badly set, as described above.

In the final experiment with the antenna trace we get a promising result. The result shows that the method can be used in a more complex environment, where the attacker does not have physical access to the device and where the timing of encryption executions is unknown. Even strong interfering signals from surrounding devices is handled by the described normalization procedure. Furthermore, as the extraction takes less time than the total recording time of the traces being processed, it is probable that extraction could be performed in real time.

B. Envelope segment size

For our experiments we use an envelope segment size of 600 samples. This value is chosen with two aspects in mind; the length of each encryption round and the length of an entire encryption block. Since the correlation array contains many lower peaks separated by the length of one encryption round, we choose an envelope segment that is long enough to cover up the gap between these peaks. At the same time, we keep the segment significantly shorter than the encryption blocks to ensure that the envelope has clear local maximums. In our case, the approximate length of an encryption block and an encryption round is 4000 samples and 330 samples respectively. This combination led to the choice of 600 samples segment length.

C. Flaws in our data

Here we list flaws in our data that we discovered during our work. Since the SDR was loaned and the time limited we did not have the opportunity to correct these mistakes. The flaws are taken into consideration when compiling the results.

1) Lost encryption blocks: To test the templates we collected the data set *For testing* by recording the side-channels during ten encryptions. To keep this data set below $30\,\mathrm{GB}$, we let the trace length be $130\ 000$ samples. This is enough to let the time between start of recording and start of encryption to vary with $\pm 8.64\,\mathrm{ms}$. Despite this we lost 111 encryption blocks in our data set *For testing* due to timing errors, giving a total of $499\ 889$ encryption blocks.

2) Deep Learning: During this project we aimed to recreate the attacks described in [4] and/or [8], combined with our proposed method. By doing so we would show if our method is actually useful in this type of attack. Unfortunately, while training the suggested model from [4] we could see that the accuracy never improved, meaning that the model did not learn. By comparing with traces from [8], we have concluded that our data lacks in detail and we believe that this made the DL network unable to learn. As our extraction of encryption blocks does not modify the data within the blocks, we suspect that the details were lost in the amplitude demodulation during data capture. We used a low-pass filter with a cut-off frequency $f_c = 1.2\,\mathrm{MHz}$, which is probably too low. This hypothesis is however not confirmed and needs further investigation.

VIII. CONCLUSION

We demonstrate that:

- cross correlation is a suitable method to locate and extract encryption activity from side-channel signals;
- the template used for correlation can be made with data from just a few encryption executions, average of more than 100 executions is excessive;
- the extraction is quick enough to be used in a real time attack.

Furthermore, we are confident that our method of extraction will work together with previous key-recovery methods.

Future work includes capturing a data set for a key-recovering deep learning network, using our suggested method. This can also be expanded into a complete side-channel attack procedure, probably able to operate in real-time. Investigating if the suggested method can be used for attacks against other cryptographic implementations is also worth considering.

REFERENCES

[1] A. Kuehlmann. (2020, Aug.) Hardware security: A critical piece of the cybersecurity puzzle. [Online]. Available: https://semiengineering.com/hardware-security-a-critical-piece-of-the-cybersecurity-puzzle

[2] A. K. Khan and H. J. Mahanta, "Side channel attacks and their mitigation techniques," in *2014 First International Conference on Automation, Control, Energy and Systems (ACES)*, 2014, pp. 1–4.

[3] G. Camurati, S. Poeplau, M. Muench, T. Hayes, and A. Francillon, "Screaming channels: When electromagnetic side channels meet radio transceivers," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 163–177.

[4] R. Wang, H. Wang, E. Dubrova, and M. Brisfors, "Advanced far field EM side-channel attack on AES," in *Proceedings of 7th ACM Cyber-Physical System Security Workshop (CPSS 2021)*, Hong Kong, China, 2021, (In press).

[5] M. Dworkin, E. Barker, J. Nechvatal, J. Foti, L. Bassham, E. Roback, and J. Dray, "Advanced Encryption Standard (AES)," Federal Inf. Process. Stds. (NIST FIPS), National Institute of Standards and Technology, Gaithersburg, MD, Tech. Rep., Nov. 2001. [Online]. Available: https://doi.org/10.6028/NIST.FIPS.197

[6] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems," in *Advances in Cryptology — CRYPTO '96*, N. Koblitz, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 104–113.

[7] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology — CRYPTO' 99*, M. Wiener, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 388–397.

[8] R. Wang, H. Wang, and E. Dubrova, "Far field EM side-channel attack on AES using deep learning," in *Proceedings of the 4th ACM Workshop on Attacks and Solutions in Hardware Security*, ser. ASHES'20. New York, NY, USA: Association for Computing Machinery, 2020, p. 35–44.

[9] Kokke. (2013) Small portable AES128/192/256 in C. [Online]. Available: https://github.com/kokke/tiny-AES-c/

Design and Development of a Communication Middleware for Distributed Embedded Systems Using Code Generation

Alex Sidén and Morgan Adamsson

***Abstract*—With the increasing need for larger and more powerful distributed embedded systems comes the need for more tools to manage them. One area where such a tool is needed is the internal communication for a distributed embedded system. This project focuses on the development of such a tool, namely the development of a communication middleware whose network configuration is automatically generated by a predefined network model. The middleware is responsible for routing information on the network and relieves the application developer of this responsibility. By using a multi-layer approach, additional functions can be easily implemented.**

The middleware demonstrated reasonable results for simple networks. However, it does not take into account the timing characteristics of the platform. A fact that currently prevents its use in most large networks.

***Sammanfattning*—I takt med att behovet av större och kraftfullare distribuerade inbyggda system ökar, ökar även behovet av verktyg för att hantera dessa system. Ett område där ett sådant verktyg behövs är den interna kommunikationen i ett distribuerade inbyggt system. Det här projektet fokuserar på att utveckla ett sådant verktyg, genom att utveckla en modellbaserat programvara som automatiskt konfigureras med kodgenerering utifrån en fördefinierad modell över nätverket. Eftersom programvaran hanterar datatransmissionen i nätverket, avlastas detta ansvar från applikationens utvecklare. Genom att använda en modellbaserat metod kan framtida funktioner enkelt implementeras.**

Programvaran visade ett rimligt resultat för enklare nätverk. Men programvaran tar inte hänsyn till plattformens egenskaper. Något som förhindrar programvaran från att tillämpas på storskaliga nätverk.

***Index Terms*—Distributed Embedded Systems, Transport Layer, Code Generation, Automatic Configuration, Communication, CAN.**

Supervisor: Matthias Becker

TRITA number: TRITA-EECS-EX-2021:200

I. Introduction

A. Background

With the ever-increasing demands on embedded systems and the need to compute a growing amount of data both faster and more accurately, comes the need for more computing power. This is often realized in the form of more, faster, and specialized Electronic Control Units (ECU). However, this increasing number of ECUs has also led to problems, with some of the larger problems being those of communication and integration.

The automotive industry is a prime example of where ECUs are used. A single car can have over 100 ECUs, each performing its own tasks such as sensing, signal processing, etc. With so many ECUs, a consistent and reliable way of communicating is necessary.

The communication problem is not a technical problem as there are already several established ways to send signals between different ECUs. However, it is a signal management problem. Adding a signal to one ECU requires writing code to transmit and receive signals on another ECU. Multiply this by thousands or tens of thousands of signals and it is easy to see that this can become quite overwhelming.

This complexity can be solved by using code generation to automatically generate code from predefined application models, but also by configuring the platform for the particular application of the middleware [1].

B. Project formulation

The goal of this project is to design and build a communication middleware for distributed embedded systems. A middleware that allows different tasks to communicate with each other on the same ECU and between different ECUs. The middleware also allows tasks to be moved freely between different ECUs during the design phase. Without the need to rewrite any application code and without affecting the functionality of the application.

To create this middleware, two things are needed. First, a model that describes how the communication middleware is built. It must be both modular and extensible to work with different communication protocols. The model needs to be detailed, describe how everything works and describe how the different parts of the middleware interact with each other. In addition, the model must describe how the middleware interacts with the tasks and the underlying hardware.

Secondly, a code generator is needed. A program that can generate multiple configuration files for each ECU. These files describe how the data is encoded and decoded. What type of data is used within the ECU and what data needs to be transferred to and from other ECUs.

These two components are then combined to create an implementation of the communication middleware. Due to the context of this thesis, the only inter-ECU communication protocol considered is CAN. The code generator provides the middleware with the necessary configurations, which in turn are extracted from an Amalthea model [2]. A model that

describes how an application is structured, on which ECU the tasks are located and what data the tasks receive and transmit.

CAN [3] is a communication protocol widely used in the automotive industry. CAN transmits data on the CAN bus using data packets called frames. Multiple devices can be connected to the same bus. Each frame has an associated data payload and address. To guarantee that only one device can communicate on the CAN bus at a time, the specification defines an arbitration scheme. Arbitration is enforced by assigning a priority to each frame, where the priority is defined by the address of the frame.

The code generator is provided with an Amalthea model [2] that describes the network. The code generator should attempt to optimize network bandwidth by mapping signal IDs to CAN frames, with the possibility of multiple signals being transmitted on the same CAN frame. Since each CAN frame has a priority associated with its address, the assignment of each frame's address must be done using an optimal priority assignment algorithm [4].

Each ECU has its own set of tasks that are executed by a real-time operating system. The application is defined as the ensemble of tasks running on each ECU. The tasks use data units called signals to exchange data. Each signal has a unique identifier (ID), and the signal data can vary in size. Because the application runs in a real-time environment, the tasks are subject to timing constraints.

An example of an existing architecture is the AUTOSAR classical model [5]. The AUTOSAR COM-stack is more extensive than required by the applications of this middleware, and is therefore not implemented. However, some inspiration has been taken from their existing specifications.

II. Proposed middleware

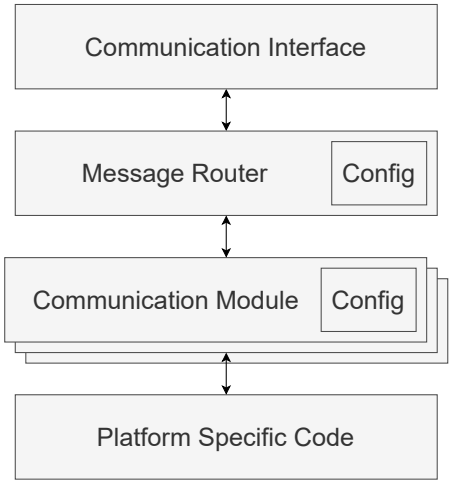


Fig. 1. A top level view of the network stack that illustrates how each layer interacts with each other.

The proposed middleware, as shown in figure 1, is a network stack inspired by the AUTOSAR COM-stack [5]. Its primary purpose is to route signals between different parts of an application that is potentially distributed across multiple ECUs.

The middleware is divided into four layers: Communication Interface, Message Router, Communication Modules, and Platform Specific Code. Each layer of the stack is designed to perform a specific task, interacting only with its neighboring layers. The message router and the communication modules also contain a configuration module that determines how data is processed and transmitted. This allows the middleware to be very flexible and work with different networks without having to adjust any middleware code. Each layer can communicate with the adjacent layers by having a well-defined interface between them.

A. Communication interface

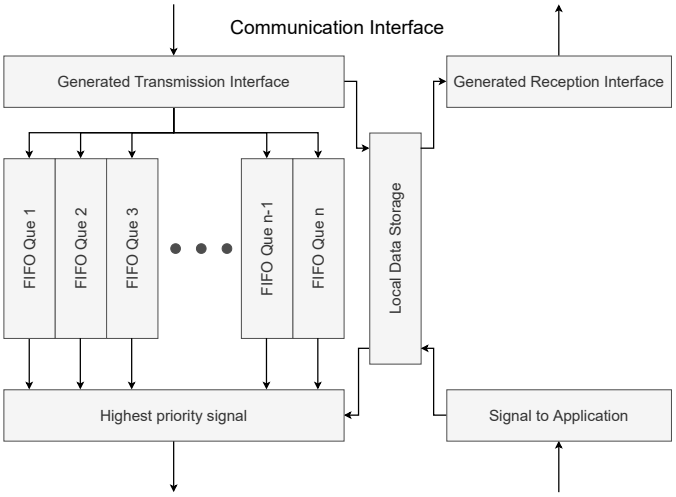


Fig. 2. Illustrates how data is flowing through the communication interface. The priority queues consist of n discrete FIFO queues with a priority associated with each respective queue.

The communication interface, as shown in figure 2, is the interface between the tasks and the communication middleware. Tasks interact with the interface by calling one of two automatically generated functions, depending on whether the task intends to transmit or receive a signal. It is worth noting that any signal transmitted by a task can be immediately received by another task, since the local data storage is shared by the transmitting and receiving functions of the interface.

Once a task sends a signal to the interface, three things happen. First, the signal's data is stored in a shared data storage. Then, a local flag is set to indicate that the signal is currently being processed on the ECU. Finally, if the signal is not currently being processed, it is placed in a priority queue, where the priority of the signal is indirectly determined by the application itself. Something that is covered in section III-A.

The priority assignment of each signal is implemented as rate-monotonic [6]. The priority queue is a set of First In First Out (FIFO), queues. The size and number of each queue depends on how many signals and priority levels are required by the application. The order of the queue represents a priority level, where a lower order corresponds to a higher priority in the priority queue. The number of queues is determined by grouping all signals into multiple sets, where each set contains all signals with the same transmission period. Here,

the number of sets corresponds to the number of FIFO queues and the number of signals within each set corresponds to the size of these queues, with a lower transmission period representing a higher priority.

Once there are one or more signals within the queue, the message router can fetch and remove the signal with the highest priority. The message router is also capable of routing signals to the communication interface. Once a signal is routed in this way, two things happen. First, the signal stores the new data in the shared data storage. Then, the local flag that was previously set in an earlier part is reset, signaling to the ECU that the signal is no longer being processed.

B. Router

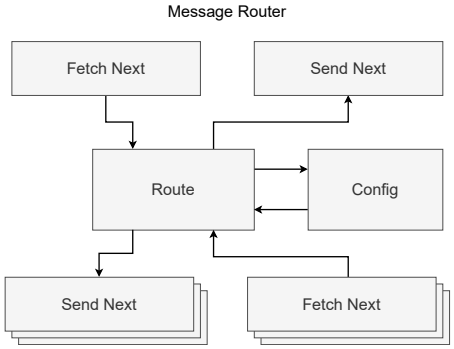


Fig. 3. An overview of how the message router operates. It communicates with its neighboring layers and uses the configuration to determine the destination of a signal.

The goal of the message router is to route arbitrary signals from any module to any other module, using a configuration as a guide. A simplified, illustrative version of this is shown in figure 3. The message router works by traversing each connected module and checking to see if that module contains a signal that needs to be routed. Once a signal is found, the router uses the configuration to determine the destination of the signal. The signal is then transmitted to its new destination module.

C. Communication modules

The proposed middleware is capable of supporting multiple communication modules with different communication protocols. However, the implementation of the proposed middleware only includes CAN. For this reason, the only implemented communication module is CAN.

The CAN module, as shown in figure 4, is a communication module that translates signals into CAN frames and CAN frames back into signals. These frames contain enough information to be understood by the CAN driver and transmitted over the CAN bus.

The CAN module is divided into a transmitting side and a receiving side. The transmitting side works by first receiving a signal ID from the router. The signal ID is used to map the signal to one or more CAN frames, depending on how the configuration is set. Once the signal is converted, the local flag mentioned in section II-A is reset. These CAN frames are expressed as generic objects that cannot be used by the hardware drivers. They are cached in a frame FIFO queue before being sent to the CAN wrapper. The CAN wrapper converts these frames into a platform-specific CAN frame that can be transmitted over the CAN bus.

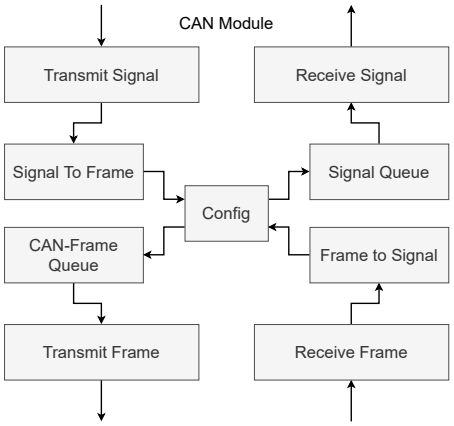


Fig. 4. Block diagram displaying how the CAN module operates. It's split in two columns that convert signals to CAN frames and CAN frames to signals.

The receiving side is a little more complicated, as it depends on the router. The process starts when the router checks if a new signal is available in the CAN module. This in turn causes the CAN module to request a new CAN frame from the hardware drivers by interacting with the CAN wrapper. If no new frames are available, nothing happens. However, if a new frame is available, the CAN module receives the frame. This frame is converted into one or more signals by the CAN configuration. Each signal sets the corresponding local flag mentioned in section II-A. These signals are cached in a signal FIFO queue before being forwarded to the router one by one when requested.

The size of both the transmitting CAN frame queue and the receiving signal queue is calculated based on the frames and signals passing through the CAN module. This is explained in more detail in section III-A.

D. Configuration

The main task of the configuration module is to connect the middleware to the network. The configuration is based on an Amalthea model [2] that lists signals with corresponding tasks. The model also specifies the mapping between ECUs and tasks and how the ECUs are physically connected.

There are two different types of configurations. The first is the router configuration, whose input is a signal ID and whose output is the destination module of the signal. If the signal is destined for the ECU, the router configuration outputs the communication interface as the destination. However, if the destination of the signal is another ECU, the router configuration outputs the communication module that will forward the signal to the destination ECU. This assumes that a signal can only have one destination ECU.

The second type is the communication module configuration. Each of the communication modules has a configuration

that is generated depending on the requirements of the network. The communication configuration can either receive a signal ID or an entire protocol frame.

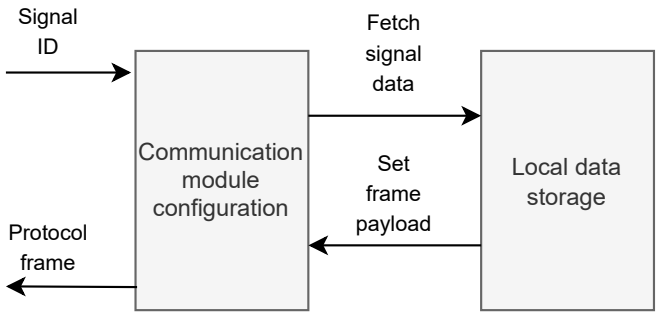


Fig. 5. Configuration module operation for transmitting a signal.

For each communication module, the signals can go in two directions. A communication module either transmits or receives a signal from the physical data bus.

Figure 5 illustrates a transaction when a signal is transmitted from ECU. First, a signal ID is sent to the configuration module, and the signal ID is matched against a protocol configuration. The configuration module retrieves the signal data from the local data storage and populates the protocol frame payload. The configured protocol frame is finally returned as output.

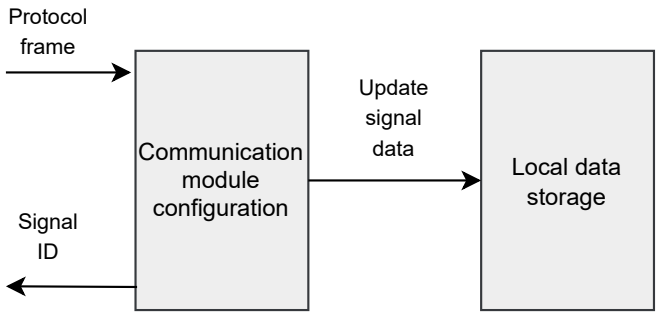


Fig. 6. Configuration module operation for receiving a protocol frame.

Figure 6 illustrates a transaction when a protocol frame is received from ECU. The protocol frame is sent to the configuration module and the corresponding frame address is matched with one or more signal IDs. The configuration module updates the corresponding signal data in local storage and returns the received signal IDs as output.

The communication configuration described above is a sample implementation of a CAN configuration. The implementation may vary depending on the protocol.

E. Specific to Platform

The only layers that are necessarily platform-specific are the communication protocol drivers and the task wrapper for the operating system.

The protocol drivers are wrapped by a well-defined interface to ensure that a communication module can interact with its corresponding driver. Preferably, the platform-specific wrapper should only receive and transmit protocol frames on the physical data bus and leave further processing to the modules further up the stack.

The platform operating system must provide each task with a suitable environment for execution and be able to preemptively schedule each task according to the timing constraints of the Amalthea model.

III. Implementation

A. Code generation

The code generator is a seemingly large and complicated piece of software that converts an Amalthea model into configuration files. In this thesis, Java is used as the main programming language for logical operations and Xpand is used to write the templates for code generation. The Amalthea model describes what signals are transmitted and received by each task, on which ECU each task is executed, timing information, and how the ECUs are physically connected.

The code generation process is divided into five steps. Each step builds on the work of the previous steps. The following list is a simplified guide to how the process works.

Step 1: The first step in the code generation process is to parse the Amalthea model. Here, each signal transmitted by each task is assigned a unique ID, priority, and transmission period based on their properties in the Amalthea model.

Step 2: The second step deals with mapping. It starts by iterating through each ECU and checks which signals each ECU receives and transmits. If a signal is both received and transmitted by the same ECU, it is marked as an internal signal and is effectively treated as a local variable on that ECU. However, if a signal is transmitted on a different ECU, the signal must be mapped on both the transmitting ECU and the receiving ECU. In this case, the signal is marked by both ECUs, informing future steps that the signals are to be transmitted by a communication module.

Step 3: The third step is the generation of CAN frames. Each signal passing through two different ECUs is assigned to one or more CAN frames. The assignment depends on the size of the signal and how often the signal needs to be sent. For example, if a signal is larger than eight bytes, the size of a CAN frame, it is fragmented into several different CAN frames. If, on the other hand, the signal is very small, it is possible that it will be packed into one CAN frame along with other signals. Once each signal on each ECU has an assigned CAN frame, each frame is assigned a unique address based on its priority. This address represents its physical priority on the CAN bus.

Step 4: In the fourth step, the buffer sizes of the communication interface, the CAN module, and the frame buffers required in the platform-specific code are calculated. The sizes of all these buffers depend on the signals that are passed through the middleware. The buffer sizes are determined by counting the maximum number of frames or signals that may be transmitted during any given time period. It is also possible to set custom buffer sizes. This is not recommended because buffers that are too small make the system unstable, and buffers that are larger than necessary waste system resources.

Step 5: The fifth step is the generation of the code. Here, the setup from each previous step is used to generate the necessary configuration files. These files are needed to make the middleware work according to the specified Amalthea model. The generated files contain C code that can be placed in the working directory of the ECU project. The generated code consists of the router configuration, CAN configuration, signal definitions, and the communication interface. The configuration code is generated as lookup tables, with each entry specifying how the middleware should process each signal.

B. Platform

The middleware was primarily developed for FreeRTOS [7] running on the STM32F105 [8] family of microcontrollers.

The CAN wrappers are written using the hardware abstraction library (HAL) [9] available for the STM32F105. The CAN frame buffer provided by the CAN drivers available in the HAL library is generally not large enough to accommodate fragmented signals or when multiple tasks attempt to send signals simultaneously. Therefore, the CAN wrapper includes a circular buffer [10] that buffers frames that would otherwise not fit in the internal CAN transmit buffers.

Conveniently, HAL provides callbacks whenever there is room in the driver's internal frame buffer. As soon as there is room to transfer a CAN frame to the internal CAN transmit buffer, the callback is invoked that takes the first CAN frame in the external circular buffer and adds it to the internal CAN transmit buffer.

One limitation of this implementation is that the CAN wrapper will throw away the sent frame if the circular buffer is not large enough. This could be a problem if the discarded frame had a payload of a fragmented signal, as this chunk will never reach the destination ECU. This could potentially prevent the receiving ECU from updating its locally stored signal, which could be disastrous in a real-time environment.

C. Integration

The communication middleware must be integrated into an embedded system for it to work. It is not a standalone program, but a complement to an overlying application. Therefore, it must be invoked by the host platform every time the platform wants the middleware to relay a signal.

The current implementation is divided into two parts, enqueuing signals and routing signals. Enqueuing works by each task executing a function that adds the signal ID to the priority queue. This is the queue seen in the communication interface in section II-A. The signal ID is stored here until routing begins.

There is no set start condition for routing signals. The application developer is responsible for initiating the message router, which can be invoked at any time. Once invoked, the router checks for signal IDs pending on the communication interface or the various communication modules. If any signals are found, they are forwarded to their destinations according to the router configuration.

One of the more common ways to integrate routing or "signal transmissions" is to run the routing as a standalone task. Where the task is executed at a predefined time period or when a set of predefined conditions are met. Something that allows it to be treated like any other part of a large embedded system.

D. Signal packing

The signals used within the middleware have different sizes, some are only one bit in size, others can be larger than the size of a single CAN frame. Since the CAN protocol itself has a relatively large overhead, each frame must contain as much data as possible. By sending only one signal per frame, most of the data transmitted over the CAN bus would consist of metadata and control bits. This results in fewer signals being transmitted on the CAN bus due to poor bandwidth utilization.

To maximize the amount of data sent per frame, some signals need to be packed into a single CAN frame. To achieve this, a packing algorithm is needed. Several frame packing algorithms already exist, some of the more efficient ones are described in [11]. However, implementing one of these efficient algorithms is problematic because the preconditions of the middleware do not match the preconditions of the algorithm. For example, one of the prerequisites of the algorithm is that each transmitted signal resides on the same ECU, which is generally not true for the middleware. Therefore, a simplified version of the [11] proposed algorithm is used for frame packing.

The packing algorithm works by first sorting the signals into lists, where each list represents each signal with a specific transmission period. The signals within each list are then sorted by size, from largest to smallest. This is followed by a bin-packing algorithm that generates a list of CAN frames and takes the first signal and tries to fit it into the first frame in the list. If the signal does not fit in, it tries to place it in the second frame and so on. Once each signal has been placed in a frame, the packing of that specific transmission period is complete. This is then repeated for each signal list, resulting in a set of packed CAN frames where each frame contains only signals with the same transmission period.

To transmit a packed frame, the CAN configuration waits for each signal within a specific frame to be updated. Then it takes those signals and packs their data as densely as possible into the frame. The largest signals are packed at the beginning of the frame and the subsequent smaller signals are placed one after the other.

Receiving a packed CAN frame is done by simply reversing the signal packing. The CAN configuration updates each signal in the local data storage, and returns the set of signals received in the packed CAN frame.

E. Fragmentation of signals

The CAN protocol can support a maximum of 8 bytes of payload per frame. To transmit larger signals over the network, the signals must be fragmented into multiple frames. The signals can then be reconstructed on the receiving ECU.

Transmitting a signal larger than 8 bytes is relatively simple. The transmitting ECU splits the signal into n chunks, where

$$n = \lceil \frac{size_{signal}}{8} \rceil \quad (1)$$

The $n-1$ first chunks are 8 bytes long, and the nth chunk is $size_{signal} \mod 8$ bytes long. Each chunk is then sent as a payload in individual frames. To simplify the implementation of packing and fragmentation, no additional signals are appended to a chunk.

Receiving a fragmented signal is a more complex process. Once the CAN configuration receives a CAN frame whose address matches the signal ID of a fragmented signal, the configuration updates the corresponding bytes of the locally buffered signal and marks that part of the signal as received. If the signal is not fully reconstructed, the configuration outputs a status code informing the CAN module that the signal is not fully reconstructed and should not be forwarded to the router.

Once all frames have been received, the configuration returns the signal ID to the CAN module, and the reconstructed signal is treated as any other signal.

If a task attempts to read the signal data while it is being reconstructed, the old signal data should be used to avoid using corrupted data. This means that an additional buffer is required for reception, with the configuration copying the contents of the receive buffer to the signal buffer once the signal has been fully reconstructed.

F. Address assignment

The address of a CAN frame determines its priority on the CAN bus, with a lower address corresponding to a higher priority. Therefore, it is important to assign the correct CAN address to a frame. An incorrect address can cause signals to arrive too late.

Address assignment begins by storing each CAN frame used by the middleware in a list. Then this list is used to assign a relative priority to each frame based on its contents, namely its signals. Each frame goes through its signals and selects the signal with the lowest transmission period. If there are multiple signals with the same period, the signal with the highest priority is selected. Consequently, the signal with the highest priority is always selected. Since the simplified signal packing discussed in section III-D only packs signals with the same period.

The selected signal is then used as the new relative priority of the frame. The frames are then sorted from highest relative priority to lowest relative priority and given an address based on the frame's position in the list. For example, the first CAN frame is given an address of 100, the second is given an address of 101, and so on.

IV. Evaluation and results

A. Setup

The middleware described in the previous section is evaluated here to see how task placement affects signal propagation time. In this experiment, a total of five tasks are used, namely task 1 to task 5. Task 1 starts by sending a signal to task 2. Task 2 modifies the signal slightly and passes it to task 3. Task 3 also modifies the signal and passes it to task 4 and so on until the modified signal reaches task 5. Task 5 also modifies the signal, but sends it back to task 1. Task 1 checks whether the signal has been modified by all other tasks and displays the round-trip time that task 1 took to receive the updated signal.

In the following experiments, each task is executed with a period of 100 ms in a predefined order. The middleware is activated whenever the ECU is idle, i.e., when no tasks are executing. This effectively means that an ECU fetches all updated signals before the first task is executed and transmits each updated signal once all tasks have been executed.

The ECUs are STM32F105 microcontrollers, running at 72 MHz. Each task is executed by FreeRTOS, which runs on each microcontroller. To ensure the order of each task, priorities are assigned to the tasks. Task 1 has the highest priority and task 5 has the lowest priority.

The round-trip time is calculated by measuring the time difference between when task 1 sends the first signal B and when task 1 receives an updated signal A. This is done using FreeRTOS software timers, where an interrupt that increments a counter is called once per millisecond.

TABLE I
SHOWS THE MAXIMUM AND MINIMUM ROUND-TRIP TIME MEASURED FOR A SIGNAL TO PROPAGATE THROUGH THE SYSTEM.

	Min round-trip time	Max round-trip time
Experiment 1	100 ms	100 ms
Experiment 2	200 ms	200 ms
Experiment 3	300 ms	300 ms
Experiment 4	300 ms	700 ms

B. Experiment 1

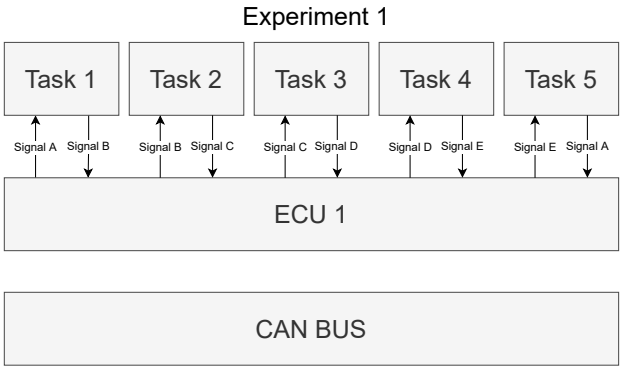


Fig. 7. Experiment 1, the tasks are executed from left to right for each ECU. Each ECU runs asynchronously once every 100 milliseconds.

The first experiment serves as a baseline. Each task is located on the same ECU and executed in order, starting with task 1 and ending with task 5, as shown in figure 7. Since the tasks reside on the same ECU, each updated signal is immediately available to every other task and no signals are transmitted over the CAN bus. This results in task 1 getting a correct result every time ECU executes the task, giving a round-trip time of 100 ms.

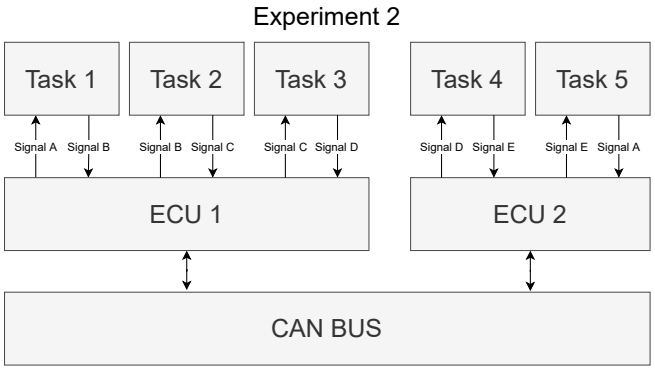


Fig. 8. Experiment 2, the tasks are executed from left to right for each ECU. Each task runs asynchronously once every 100 milliseconds.

C. Experiment 2

In the second experiment, the tasks are split between two different ECUs. ECU 1 executes task 1 through task 3 in order, and ECU 2 executes task 4 through task 5, as shown in figure 8. Since the tasks are now executed on different ECUs, the signals must be transmitted via the CAN bus.

D. Experiment 3

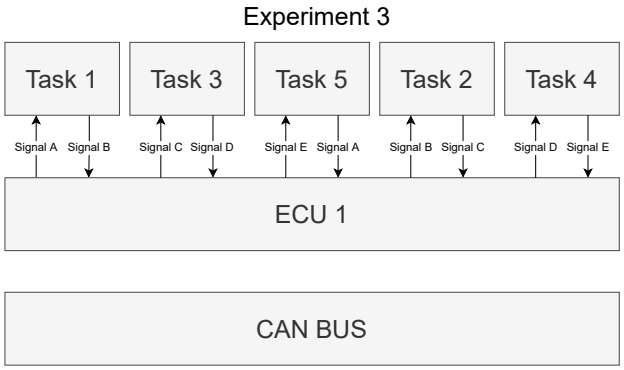


Fig. 9. Experiment 3, the tasks are executed from left to right for each ECU. Each ECU runs asynchronously once every 100 milliseconds.

The third experiment aims to see how the order of task execution affects the outcome. As in experiment 1, all tasks are on the same ECU, but the tasks themselves are executed in the following order: task 1, task 3, task 5, task 2 followed by task 4, as seen in figure 9. As the execution order is changed, the round-trip time required for task 1 to get the correct result also changes. This results in a round trip time of 300 ms.

E. Experiment 4

In the fourth and final experiment, the tasks are again split between two different ECUs. ECU 1 executes the following tasks in the following order: task 1, task 3, and task 5. ECU 2 executes the following tasks in the order of task 2 followed by task 4, as shown in figure 10. As in experiment 2, some signals need to be transmitted over the CAN bus. Interestingly, the round-trip time varies between 300 ms and 700 ms, which is a result not found in any of the previous experiments.

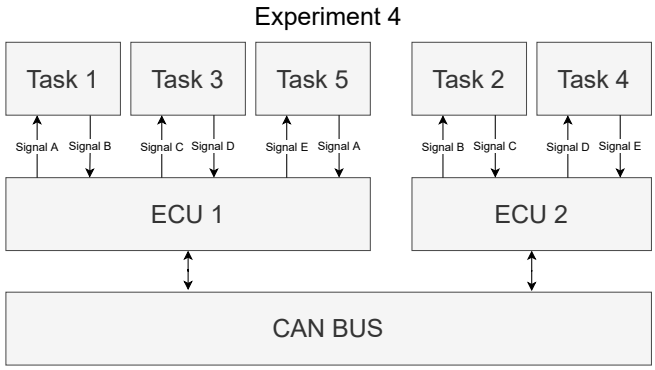


Fig. 10. Experiment 4, the tasks are executed from left to right for each ECU. Each ECU runs asynchronously once every 100 milliseconds.

V. Discussion

A. Evaluation

In the experiments, signal propagation time was measured for an application running on either one or two ECUs. The measured difference becomes clear when comparing the results of experiment 1 with experiment 2 and experiment 3 with experiment 4. The results of the experiments can be found in the table I.

The first comparison is between the results of experiment 1 and experiment 2. It is easy to see that experiment 2, running on multiple ECUs, has an overall larger round-trip time compared to experiment 1. This is somewhat unexpected, since the application in experiment 2 should have a minimum round-trip time of 100 ms, just like in experiment 1. The most likely explanation is that ECU 1 and ECU 2 are, by chance, executing their tasks simultaneously. This means that ECU 2 executes task 4 before ECU 1 can transmit signal D, and that ECU 1 executes task 1 before ECU 2 can transmit signal A.

The second comparison, between the results of experiment 3 and experiment 4, shows that experiment 4 has the potential to have the same round-trip time as experiment 3. This is excellent because it demonstrates the potential of the middleware to distribute an application across multiple control devices without affecting the application. The key word here is ”potential” because the maximum round-trip time of experiment 4 is more than twice that of experiment 3, which is most likely explained by the fact that both ECUs execute their tasks simultaneously, similar to the conclusion of the first comparison.

Both comparisons show that an application distributed over multiple ECUs suffers from increased signal propagation time. This increased signal propagation time appears to vary depending on when the ECUs execute their tasks relative to each other. It should be noted that the distributed version did not affect the functionality of the application. This indicates that any application can become distributed using the middleware.

However, this would likely not be the case for large networks, as such an application would have more moving parts. Tasks would have the potential to have different execution times. This could result in signals being sent and received at different times, perhaps even too late to be useful. This is

one reason why the middleware and the distributed embedded system as a whole should be subjected to a timing analysis. An analysis that attempts to find problems in the embedded system and calculate the best-case and worst-case latencies for each part of the distributed embedded system. Because without it, as the results show, different parts of the application can become desynchronized, resulting in an application that works slower than planned or not at all.

This simply means that the communication middleware is not a one-step solution for building a distributed embedded system. It is, however, a good starting point. This is because the middleware gives developers the ability to build distributed embedded systems without having to worry about signal management. In return, other system designs must be used that can compensate for the weaknesses of the middleware. Future designs for embedded systems may even have the potential to negate the shortcomings and make embedded systems even more distributed.

For example, by designing a distributed embedded system that stores a timestamp in each signal and a signal history in each task. It would be possible to approximate a lost or delayed signal by using previous signal values. This design could compensate for the middleware problems, but requires predictable signals.

B. Timing Analysis

There are several factors that determine whether or not a network is schedulable. Primarily, signal priorities and CAN addresses determine schedulability, but factors such as transmission speed on the data bus also play a detrimental role. Since data bus speed, execution time, execution time variance, or task execution order are not available, no specific timing analysis was performed in this thesis.

In addition, some basic analysis such as overhead or round-trip latency in milliseconds can and should be evaluated for the middleware. However, due to time constraints, these measurements were neglected.

C. Future Development

There are many ways in which the middleware can be either extended or improved. One such enhancement, as mentioned earlier, is the addition of additional communication modules to support more complex system configurations. Other enhancements include adding ”nice to have” functionality, such as evaluating generated system properties and determining whether the system is plausible. Three areas of enhancement are discussed in the following paragraphs.

The proposed stacked layer model developed for the middleware is one area of enhancement. It was designed with several features in mind that were not found to be useful in the developed middleware. For example, the middleware should be able to act as a gateway for multiple communication protocols and be more tightly integrated with the platform it runs on to reduce overhead. Implementing these things along with multiple communication protocols would allow the middleware to be applied to more diverse distributed embedded systems.

The code generator and subsequent implementation of a new configuration is also an area that can be improved. Here the code generator could be improved to support different configurations of the Amalthea model and use more advanced mapping methods. This would allow the code generator to improve the way each signal is routed. Such improvements are desirable when the Amalthea model requires multiple communication protocols or when an ECU acts as a gateway.

The work environment itself is one area that can be improved. Interestingly, this improvement is a thesis proposal, namely ”From Application Model to Implementation - Generating Application Skeletons for Real-Time Operating Systems”. The ability to generate code skeletons for any platform would have simplified the middleware development. It would also allow the middleware to be ”attached” so that each new code skeleton could be pre-packaged with a generated middleware.

VI. Conclusion

The stacked-layer approach in this project proved to be very useful in the development of the middleware as it isolated the development and function of each layer. This allowed for easy implementation of additional functionality.

The code generation successfully managed to configure each ECU to transmit signals according to the network model. However, the code generation did not take any precautions regarding the schedulability of the network.

The middleware has proven capable of handling data transmission in simpler networks, but assumes that each network can already handle varying transmission times and delays. Something that currently prevents it from being used in most large scale networks.

Acknowledgment

The authors would like to thank the thesis supervisor Matthias Becker for his support and very useful feedback during the development of the project.

References

[1] K. Goseva-Popstojanova, T. Kahsai, M. Knudson, T. Kyanko, N. Nkwocha, and J. Schumann, “Survey on model-based software engineering and auto-generated code,” NASA, Tech. Rep. NASA/TM–2016–219443, Oct 2016.

[2] (2021, Apr.) Eclipse app4mc. APP4MC. Amalthea. [Online]. Available: https://www.eclipse.org/app4mc/

[3] *CAN Specification*, 2nd ed., Bosch, Wiener Straße, 70469, Stuttgart, Germany, Sep 1991. [Online]. Available: https://www.kvaser.com/software/7330130980914/V1/can2spec.pdf

[4] R. Davis, A. Burns, R. Bril, and J. Lukkien, “Controller area network (can) schedulability analysis: Refuted, revisited and revised,” *Real-Time Systems*, vol. 35, pp. 239–272, Feb 2007.

[5] (2021, Mar.) Classic platform. AUTOSAR. Theresienhoehe 30, 80339 Munich, Germany. Model. [Online]. Available: https://www.autosar.org/standards/classic-platform/

[6] C. L. Liu and J. W. Layland, “Scheduling algorithms for multiprogramming in a hard-real-time environment,” *Journal of the ACM (JACM)*, vol. 20, no. 1, pp. 46–61, 1973.

[7] (2021, Apr.) Freertos kernel developer docs. FreeRTOS. FreeRTOS. [Online]. Available: https://www.freertos.org/features.html

[8] *Connectivity line, ARM®-based 32-bit MCU with 64/256 KB Flash, USBOTG, Ethernet, 10 timers, 2 CANs, 2 ADCs, 14 communication interfaces*, 10th ed., STMicroelectronics, Chemin du Champ-des-Filles 39, Plan-les-Ouates, 1228, Geneva, Switzerland, Mar 2017. [Online]. Available: https://www.st.com/resource/en/datasheet/stm32f105r8.pdf

[9] *Description of STM32F4 HAL and low-layer drivers*, 6th ed., STMicroelectronics, Chemin du Champ-des-Filles 39, Plan-les-Ouates, 1228, Geneva, Switzerland, Jul 2020. [Online]. Available: https://www.st.com/resource/en/user_manual/dm00105879-description-of-stm32f4-hal-and-ll-drivers-stmicroelectronics.pdf

[10] (2014, May) Implementing circular buffer in c. Embed Journal. Circular Buffer. [Online]. Available: https://embedjournal.com/implementing-circular-buffer-embedded-c/

[11] G. Urul, "A frame packing method to improve the schedulability on can and can-fd," Master's thesis, The Graduate School of Natural and Applied Sciences of Middle East Technical University, 06800 Çankaya/Ankara, Turkiet, Feb 2015. [Online]. Available: https://etd.lib.metu.edu.tr/upload/12618552/index.pdf

Telemetry System for Real-Time Monitoring of a Formula Student Electric Vehicle

Simon Richter and Joachim Larsson

***Abstract*—Real-time monitoring of vehicle data during testing can drastically cut down on test times as well as improve the quality of testing by facilitating the implementation of run-time compliance verification with expected model behavior, along with anomaly detection in both hardware and software. By providing a wireless communication link between vehicles and a monitoring base station, this project aims to build the groundwork for more sophisticated testing proceedings in the future. The wireless communication system implemented in this project mirrors data from the two CAN data busses on the vehicle and transmits them via a licence-free 868 MHz ISM band. The receiver is connected to a computer where the data can be visualized and analyzed in real-time. The project goals were exceeded in both throughput and range. Early testing has shown that data rates of 150 kbit/s and ranges 1.2 km and beyond are achievable. The project has set a solid foundation upon which wireless testing routines can now be developed. Hardware and software developed in this project can be built upon and optimized further in future revisions to achieve even higher data rates and longer ranges.**

***Sammanfattning*—Övervakning av fordonsdata i realtid under testprocessen kan drastiskt dra ner på testtiden samt förbättra kvalitén av testerna genom att öppna upp möjligheten för verifiering av både hårdvara och mjukvara under körning. Genom att skapa en trådlös kommunikationslänk mellan fordon och en övervakande basstation siktar det här projektet mot att lägga grunden för mer sofistikerade testmöjligheter i framtiden. Den implementerade trådlösa kommunikationslänken speglar data från fordonets två CAN-databussar och sänder de över etern till en radiomottagare. Sändningen sker via ett licensfritt 868 MHz ISM band. På mottagarsidan kan datan sen visualiseras och analyseras på en dator kopplad till mottagaren. Projektets mål har överskridits både i datahastighet och räckvidd. Tidiga tester har visat att datahastigheter på 150 kbit/s samt räckvidder på över 1.2 km går att uppnå. Projektet har lagt en stabil grund för hur trådlös testrutin kan implementeras. Hårdvaran och mjukvaran utvecklad i detta projekt kan byggas på och optimeras ytterligare för framtida revisioner. Detta kan öppna upp för ännu högre datahastigheter och räckvidder.**

***Index Terms*—Telemetry, RF, CAN, 868-GHz radio, PCB design.**

Supervisors: Mark Smith, Carl-Mikael Zetterling

TRITA number: TRITA-EECS-EX-2021:201

Introduction

With the rapid increase of complexity in modern vehicles in the form of sensors and electronic systems, collecting and analyzing sensory data in real-time has become vital both for vehicle safety, model optimization, and testing. This trend of heavy digitalization of vehicles is equally apparent within the student organization KTH Formula Student, which in its mission to build a fully driverless electric vehicle has seen the complexity of their cars increase each year. Before this project, there has not been an efficient way to monitor the different systems and sensors in the car remotely in real-time. In this bachelor's project cooperation between KTH and KTH Formula Student, a telemetry system for that purpose is designed, implemented and evaluated.

Prior to this project numerous telemetry systems have been implemented and evaluated. A common protocol for short range telemetry systems is Bluetooth since it is able to handle higher data rates with low latency. For example, a short range telemetry system was implemented on an autonomous drone by Anand M in [1]. However, in that project, a pre-made Bluetooth module was used. The focus was not on the hardware implementation of the system but rather the interface between the modules and the drone. Furthermore, a Raspberry Pi was used for more advanced data processing. In our project however, the telemetry system is built from a PCB (printed circuit board) level and interfaced with a car. Additionally, a less sophisticated ARM-based processor is deemed to be sufficient for this project. The telemetry system developed in this report uses sub 1 GHz radio frequency. Many different radio systems that used the sub 1 GHz frequencies have also been built and investigated before. LoRa is one technology that is used for transmissions requiring low power consumption and long range. An example of a LoRa based system is described by Van Torre in [2]. With an output power of only 14 dBm the described system is able to maintain a reliable communication during a distance of over 500m. The maximum range that was achieved during static conditions was 1.44 km. Since higher data rates than what LoRa can offer were needed for this project other methods were investigated and pursued.

List of acronyms

PCB Printed circuit board
RF Radio frequency
CAN Control Area Network (Communication protocol)
SPI Serial Peripheral Interface (Communication protocol)
LNA Low noise amplifier
PA Power amplifier
LOS Line of sight
FSK Frequency shift keying (Modulation type)
IC Integrated circuit
MCU Microcontroller unit
FIFO First in first out (Register type)
RSSI Received signal strength indication
HGM High gain mode

I. Problem formulation

The formula vehicle has two internal networks where sensory data is being collected and passed between different subsystems. These networks are CAN-busses. The two CAN-busses have a combined throughput of 1.5 Mbit/s. The goal of this project is to build a telemetry system that can deliver some of this data to a user in real-time while the car is moving during testing. An overview of the system is seen in figure 1.

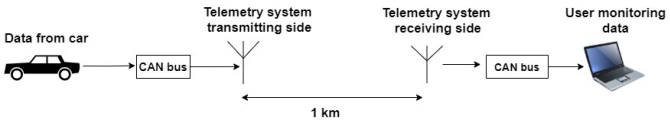


Fig. 1. Overview of the system

Roughly estimated it should suffice to have access to only a third of all signals on the busses.

Additionally will be read out by a human on a display, one tenth of the frequency of data will suffice. For that reason, 50 kbit/s was set as the target data rate in this project. The desired range for the telemetry system should be at least 1 km LOS. This would be sufficient for covering most of the test tracks on which the vehicle will be tested. The system should also be small enough to be mounted on the car in a convenient way. It also has to follow the requirements for license-free radio transmitters on the 869.4- 869.65 MHz frequency band [3]. A full set of system specifications are presented in table I below.

TABLE I
System specifications

Output power	27 dBm
Center frequency	869.4-869.45 MHz
Frequency bandwidth	50 KHz
Line-of-sight range	1000 m
Power consumption	1500 mW
f Data rate	50 kbit/s

II. Evaluation of approaches

Three different approaches to implement the telemetry system were investigated, namely: using existing cellular networks, building a WiFi network using range extenders and lastly designing a custom sub-1-GHz radio link. In the end, the custom radio link was the chosen approach for this project. A detailed evaluation and conclusion about the different approaches follow in this chapter.

1) Cellular network: One potential solution is based on using existing cellular networks (LTE, 3G, 4G) for data transfer. This may be the simplest solution to implement as it could be done using only one transmitter that uploads data to the internet, which then can be read from anywhere using a web interface. The big advantage of this solution is that the range would effectively be infinite, as long as coverage is available. The data rate of this solution would most likely also suffice the goal of 50 kbit/s even during bad coverage. The big disadvantage with this solution is the need to purchase one or two sim cards and continuously pay for data subscriptions or fixed rate data sizes to facilitate the communication. This will be both economically unviable for the team and inconvenient to maintain. Another drawback is coverage when testing the car in other countries than Sweden. There would also have to be enough motivation and knowledge within the team to constantly maintain and purchase new data during all of the years the system will be in use.

2) WiFi network with range extenders: A second potential solution is to set up a WiFi network using a base station and numerous range extenders to cover the desired testing area. This would give extremely impressive data rates exceeding the max combined Can bus speed of 1.5 Mbit/s by many ten folds. Such data rates would open up the possibility to add more data rate hungry features, such as real-time video streaming from the car. Such a solution would require several range extenders to be scattered across the testing track to offer sufficient coverage over the targeted 1km range. Common range extenders can extend the range by around 100 meters at a non-omnidirectional angle, requiring 10s of range extenders for total coverage of the test track. This would add up to a large up-front cost. Transportation and setup of such a system would also be cumbersome as the range extenders would have to be transported to - and from the testing site and be set up on site each time by someone who is familiar with the system.

3) Custom 869 Radio link: The last proposed solution is a custom radio link operating in one of the license-free ISM bands, at either 433 or 868 MHz. Many transceiver and RF front end chips exist that according to some evaluations would meet the targeted range and data rate goals. The most promising frequency band seems to be the 869.4-869.65 MHz ISM band, which allows for a highly effective radiated power at 500 mW [3]. The modulation technique of such transceivers has also been evaluated with regard to the relatively long range requirement. Mainly two techniques have been investigated, LoRa and variants of FSK. LoRa offers low data rates, mostly below 50 kbit/s. This modulation also offers very long ranges of up to several kilometers due to a very low sensitivity [2]. It was deemed that a range this long was far too much for the requirements of this project. FSK and variants of it were deemed a better fit since they offer a higher degree of freedom in terms of data rate and range. Therefore, making it easier to adjust the system to meet the requirements with higher accuracy. This solution was also deemed to be the most economical due to the low purchase price of its components and no significant cost of maintenance. Lastly, the setup of such a solution would be comparatively easy and should ideally as easy as switching on the devices with the press of a button and configuring what data is to be sent.

III. Theory

A. Link budget

Equation 1 roughly describes the delta between the transmitted signal power and the weakest signal that can be detected by the receiver. This is a useful equation for getting an idea of how much of the signal strength can be lost due to path

loss, connector losses or component mismatches.

$$Linkbudget = P_{TX} + G_{TX} + G_{RX} - S_{RX} \quad (1)$$

Where P_{RX} is the transmitted signal strength, G_{RX} and G_{TX} are the antenna gains for the transmitter and receiver respectively. S_{RX} is the sensitivity of the receiver. The sensitivity is defined as the weakest signal a receiver can detect. All units above are expressed in decibel.

Using equation 1 with a 27 dBm transmit power, an antenna gain of -2.5 dBi as in a common dipole antenna such as [4] and a sensitivity of -109 dBm as what a common sub-1 GHz transceiver such as the CC1200 [5] offers, a total link budget of 131 dBm is obtained.

B. Calculating path losses

In ideal conditions with LOS Friis Equation described in equation 2 gives a value for the received signal strength.

$$P_{RX} = P_{TX} + G_{RX} + G_{RX} + 20log_{10}(\frac{c^2}{4\pi R f_c}) \quad (2)$$

f_c refers to the frequency of the signal, c to the speed of light and R to the distance between transmitter and receiver. Identically to equation 1 G_{RX} and G_{TX} are the antenna gains for the transmitter and receiver respectively and P_{TX} the transmitted signal strength.

A more realistic way of modeling the received signal strength is using the 2-Ray ground reflection model. As the name suggests, this also takes into account the path of reflected transmissions. The total received power can with this be modeled as the sum of the direct transmission and one of the ground reflected signals. This model, however, is not perfect either as it only takes into consideration one reflection on the ground ignoring the effects of reflection from other nearby objects. Using this method should give a more realistic picture of the received signal strength than the Friis model as sources for reflection are almost always present in real conditions.

C. Impedance matched traces

In addition to matching input- and source impedance of the different RF ICs via passive component matching network to minimize reflection and maximize transferred power, it is also important to match the trace width on the PCB to 50 Ω for the same purpose. Traces on the top layer separated from a solid ground plane by a dielectric can be modeled as a microstrip transmission line as shown in figure 2.

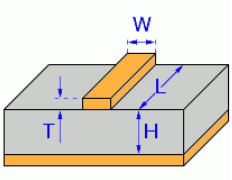


Fig. 2. Dimensions of a microstrip. Picture from KiCad EDA.

For $\frac{W}{H} \geq 1$ the set of equations 3 and 4 below proposed by Hammerstad in [6] can be used for approximating the impedance and effective dielectric constants of a microstrip trace.

$$\epsilon_{eff} = \frac{\epsilon_R + 1}{2} + \frac{\epsilon_R - 1}{2}\left[\frac{1}{\sqrt{1 + 12\frac{H}{W}}} + 0.04(1 - \frac{H}{W})^2\right] \quad (3)$$

$$Z_0 = \frac{120\pi}{\sqrt{\epsilon_{eff}}[\frac{W}{H} + 1.393 + \frac{2}{3}ln(\frac{W}{H} + 1.444)]} \quad (4)$$

H refers to the height of the dielectric material and W is the width of the trace. ϵ_{eff} is the effective permittivity and ϵ_R is the relative permittivity. Z_0 is the impedance of the trace.

IV. Methodology

A. Range estimation

The calculations for range estimations for evaluating the sub 1-GHz radio solution used a tool provided by Texas instruments. This uses both the Friis and 2-ray model presented in the theory section. For this estimation, noise and interference from adjacent channels were ignored. Instead, the model only aims to give an indication of the range achievable in ideal conditions, with the power of the transmitted signal and sensitivity of the receiver as the bottlenecks. The settings used, including sensitivity, carrier frequency, antenna gains, transmitted output power and surface conditions are shown in table II. The output power and sensitivity were obtained from the datasheets for the Texas Instruments CC1200 [5] section 4.10.2 and CC1190 [7] page 3. The antenna gain was obtained from a Wurth-Electronics dipole antenna [4].

The result of the range estimation is presented in figure 3. Without interference, this gives an estimated range of 1761 meters. This result takes the first distance at which the path losses calculated both from the Friis or the 2-Ray ground reflection model together with other losses and gains in the system go below the sensitivity of the selected transceiver. In terms of link budget, this is when equation 1 intersects zero. For the settings used the 2-ray model gave the lowest distance to give a received signal strength under the sensitivity of the transceiver. As can be seen in the same figure, the Friis equations give an intersect point around 15 km. This result is not realistic and can be discarded, as the maximum LOS that is achieved with antennas 1 meter above ground is limited by the curvature of the earth at around 7 kilometers.

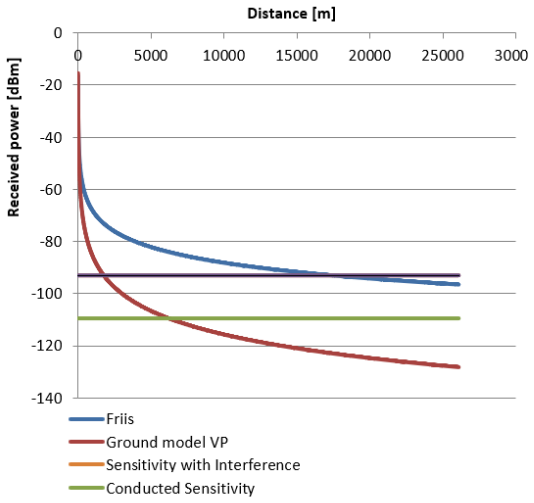


Fig. 3. A graph with the result of the range estimations, comparing a Friis-based model with a 2-Ray ground reflection model. Graph from Texas Instruments range estimator [8]

TABLE II
SETTINGS USED FOR RANGE ESTIMATION

TX antenna height over surface	1 m
RX antenna height over surface	1 m
Relative permitivity of ground (ϵ_r)	18
Mean Effect gain of TX antenna	-2 dB fixed gain
Mean Effect gain of RX antenna	-2 dB fixed gain
TX conducted output power	27 dBm
Conducted sensitivity level	-109.5 dBm
Jammer/Interference (at antenna port)	No interference

B. Hardware design

1) Hardware system architecture: The hardware architecture mainly consists of two parts: a digital section to interface with the CAN bus on the car and to interface with the transceiver, and an RF section.

The microcontroller used is a 32-Bit Arm-based chip by STMicroelectronics. With two separate CAN transceiver ICs, it acts as an interface to the car. The RF section is based on a reference design by Texas Instruments [9] using a combination of the CC1200 and CC1190 ICs. While the reference design uses a CC1120 instead of CC1200, they can be interchanged with some minor adjustments and are pin-compatible. The different sections can be seen on a render of the final PCB in figure 4.

2) Selection of hardware: To achieve the desired output power of 27 dBm it was decided that a separate transceiver and RF front end IC were to be used. The primary factor for choosing these ICs was sensitivity. Selectivity and blocking were both secondary in the evaluation of transceivers, as the test track where the telemetry unit would be used was deemed to be relatively low noise both in adjacent channels and other bands. Another factor that weighed in on the choice of RF components was ease of usage and ease of design. A comparison of ICs can be seen in the table III below.

Due to its superior sensitivity and simple integration between transceiver and RF front end, Texas-Instruments

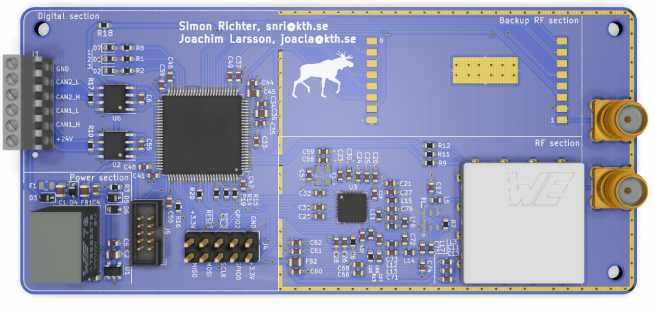


Fig. 4. A render of the final PCB.

TABLE III
COMPARISON OF EVALUATED TRANSCEIVERS

Transceiver	Data rate	Output power (dBm)	Sensitivity (dBm)
CC1200	1250 kbit/s	16 (27 with CC1190)	-109 (at 50 kbit/s)
CC1101	600 kbit/s	12 (20 with CC1190)	-104 (at 38.4 kbit/s)
CC1125	200 kbit/s	18 (27 with CC1190)	-107 (at 50 kbit/s)
CC1101	600 kbit/s	12 (20 with CC1190)	-109 (at 50 kbit/s)
CC1201	1250 kbit/s	16 (27 with CC1190)	-109 (at 50 kbit/s)
SX1272	300 kbit/s	20	-110 (at 38.4 kbit/s)

CC1200 and CC1190 were used. At the target data rate of 50 kbit/s, this offered a sensitivity of -112 dBm from the transceiver plus a typical improvement of 6 dBm from the LNA integrated in the RF front end [7]. Another factor that weighed in was the detailed documentation and excellent software tools provided by Texas Instruments to make the design process easier.

For the antenna, a Hyperion-I dipole antenna by Wurth-Electronics was used [4]. A dipole antenna was chosen rather than a monopole as it is more suitable for applications like this project, where no large ground plane is available. The chosen antenna offers a maximum gain of -2.3 dBi and a Voltage-standing-wave-ratio less than 2. As seen in figure 5 a dipole antenna offers a fixed gain in the horizontal plane when the antenna is directed vertically, which is suitable for this project as the test track where the system will be used is relatively flat.

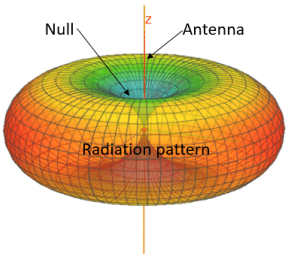


Fig. 5. Radiation pattern of a typical dipole antenna. Image courtesy of mpantenna.com

3) Digital section: The digital section is responsible for interfacing both with the car via the two CAN busses as well as with the RF link via SPI communication to the transceiver. To handle this workload a STM32F04 microcontroller is used. The microcontroller contains both a CAN and SPI controller for simple communications with the other ICs. The MCU is also connected to the transceiver via GPIO pins to

allow support for additional readings, such as the temperature of the transceiver. To translate the differential signal of the CAN bus to a single-ended signal, that can be interpreted by the microcontroller, two SN65HVD233 CAN transceivers are used. Additionally to simplify debugging and determine the state of the radio three LEDs are controlled by the MCU.

4) RF section: As previously mentioned the RF section consists of an 868 MHz Transceiver and an RF front end incorporating a PA and LNA for an output signal strength of up to 27 dBm. The transceiver uses a differential LNA while the CC1190 RF Front end outputs a single-ended signal. Therefore, in addition to the impedance matching between the output- and input of the two ICs, a balun is implemented to convert between the single-ended and differential signals. This is seen in figure 6. According to section 4.10.1 in the datasheet for the transceiver [5], while receiving it operates optimally with an output differential impedance of 60 + 60j Ω in the 868 MHz band. When transmitting the optimum load impedance in the same band is 35 + 35j Ω.

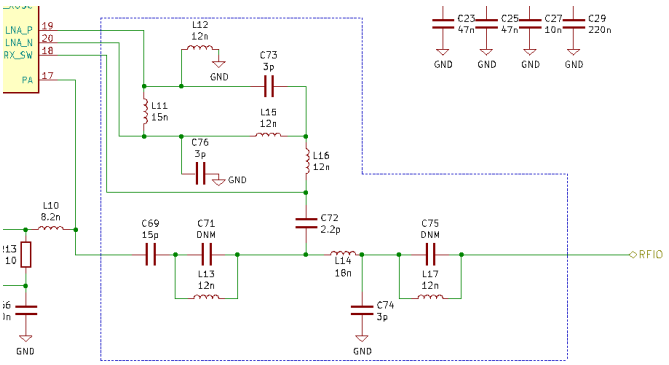


Fig. 6. The passive components making up the balun and transceiver matching. Relevant components are marked in blue dotted line.

A matching network is also present between the antenna and the RF front end to transform the RF Front ends input impedance to 50 Ω, as to match the impedance of the antenna at the SMA port.

Between the transceiver and RF-front end, a SAW band-pass filter with a center frequency of 869 MHz was used. This was recommended by the reference design to attenuate phase noise and pass regulations.

5) PCB design: The PCB uses a 4-layer stackup. The top layer contains a majority of the signal traces, as well as a ground copper pour. The first inner layer contains a large, uninterrupted ground pour covering the whole PCB. The second inner layer is a power plane. To isolate noise between the devices, the plane is divided into three separated power plains for the digital/power section, the transceiver, and the RF front-end. The bottom layer contains another large, mostly uninterrupted ground pour. Some miscellaneous digital signal routing is done on this layer. The exact PCB stackup with layer heights and dielectric properties is described in figure 7 below.

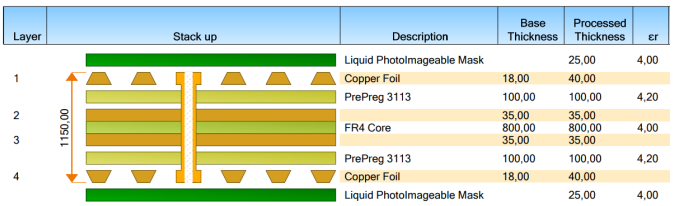


Fig. 7. Stackup of the PCB. Figure courtesy of the PCB manufacturer NCAB Group. All distances are in micrometers.

The PCB is distinctly divided into sections separating high-speed digital signals, analog RF signals and power lines from one another to minimize cross-talk and coupling between them.

Component sizes were adjusted to size 0603 (imperial) from the reference design [9] to allow for easier hand-soldering. In critical parts of the circuit where component values have to be precise, such as the matching and balun, tight tolerance inductors and ceramic capacitors were used.

At points where a 50 Ω impedance is required, the trace width of the RF signal traces has been adjusted to achieve this impedance at 870 MHz. This is mainly between the transceiver and RF front end, as well as between the RF front end and antenna. Using the PCB stackup described in appendix A with equations 3 and 4 the trace width was calculated to 200 μm. Other signal traces have a width of 250 μm. Higher current traces, in the power supply section, have a trace width of 400 μm.

C. Software

In order to operate and communicate with the different ICs on the PCB, custom software was developed for the microcontroller. The main purpose of the microcontroller is to act as an interface between the radio link and the CAN-busses. This is done differently depending on if it is the sending or receiving end of the radio link. For the sending side, the microcontroller has to extract and filter out relevant information from the CAN-buses and send it to the RF electronics to be transmitted on the radio link. For the receiving side, the microcontroller has to extract all the data from the radio transceiver and send it over the CAN bus located on the receiving end. For both sides, the microcontroller also has to configure the settings and operate the radio transceiver when the system is running. The software for the microcontroller was written in C using STM32CubeIDE [10]. The code for configuring the different peripherals of the microcontroller (CAN, SPI, Digital I/O) was generated from a tool in the STM32CubeIDE. To operate the peripherals different functions included in the HAL library were used. HAL-library is a library in STM32CubeIDE for interacting with the different peripherals in the microcontroller. The microcontroller has to communicate with three types of ICs: The two CAN transceivers, the RF front end chip and the radio transceiver.

1) CAN Interface: The goal is to mirror the two CAN-buses on the car to a computer on the receiving end of the radio link. This means that the interface between the radio transceiver and CAN-bus can be implemented easily. The

microcontroller only needs to read the two CAN-buses, pick out the desired data, and pass it on to the transceiver to be sent over the radio link. It does not have to process that data in any way, only deliver it to the transceiver. However, as described in the introduction, the radio link is made for 50 kbit/s and cannot handle the combined data rate of 1.5 Mbit/s that is on the car's two CAN-buses. Therefore the microcontroller on the transmitting end has to filter the data at a rate so that only 50 kbit/s from those 1.5 Mbit/s is sent over the radio link.

2) RF front end interface: The Rf front end is controlled via digital outputs on the microcontroller. There are three signals: Power Amplifier Enable (PA_EN), Low-noise Amplifier Enable (LNA_EN) and High Gain Mode (HGM). PA_EN activates the power amplifier in the RF front end that is used for increasing the strength of the output signal on the sending end of the radio link. LNA_EN activates the low noise amplifier used to increase the incoming signal strength on the receiving end. HGM is used to set the mode of the RF front end to one of two modes. One is optimized for output powers over +23 dBm, while the other is optimized for output powers under +23 dBm. Since the radio link communication is one-way there is no need to have LNA and PA enabled at both sides, and some power can be saved by disabling them accordingly. High gain mode is enabled on both sides since the telemetry system will primarily operate at maximum output power. The configurations for the different outputs are shown in table IV.

TABLE IV
RF FRONT END CONFIGURATION

Side	HGM	PA_EN	LNA_EN
Receiver	Enabled	Disabled	Enabled
Transmitter	Enabled	Enabled	Disabled

3) Radio transceiver configuration: The radio transceiver has 180 8-bit configuration registers that can be set to different values to adjust the settings for the radio. Some radio parameters that can be adjusted via the registers are the data rate, carrier frequency, output power and modulation type. The values for the configuration registers are written from the microcontroller to the transceiver through the serial peripheral interface communication protocol. To generate the register configuration the software tool SmartRF was used. SmartRF is a graphical user interface developed by Texas Instruments for testing, evaluating and configuring their different RF products [11]. The register values generated by SmartRF can then be exported to the microcontroller, which then transfers them via SPI to the transceiver.

4) Radio transceiver operation: The behavior of the transceiver on the transmitting end are described by a state diagram in figure 8.

On the transmitting side, the microcontroller will stay in an idle state until it has data to transmit. Once the microcontroller has data to send it will ask the transceiver if it is ready to transmit. If the transceiver is not ready it

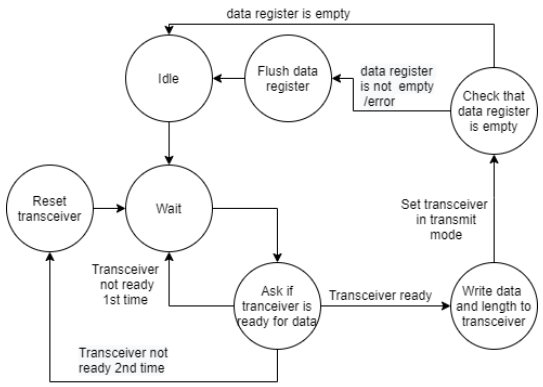


Fig. 8. State diagram of microcontroller thats operating the radio transceiver on the transmitting side

will wait for a certain time and ask again. If it still does not get a confirmation that the transceiver is ready it means that an error has occurred. If that is the case the microcontroller simply sends a reset command to the transceiver and asks again. Once the microcontroller has received confirmation from the transceiver it will first send a byte containing the length of the data to be sent. It then proceeds to send the data that it wishes to transmit. The microcontroller will then send a command for the transceiver to enter transmit mode. In this mode, the transceiver will check the length byte and put that many bytes from the data FIFO register to one packet. This packet will then be transmitted over the radio link. Once the packet has been transmitted the transceiver will automatically enter an idle state. However, if the length byte had a higher value than the number of bytes in the data FIFO register it will enter an underflow state. If the length byte had a lower value than the number of bytes in the data FIFO register it will send those bytes and still enter idle mode. Nevertheless, there will still be bytes left in the register that could be interpreted as the length byte for the next transmission. Therefore, the microcontroller has to check that the data FIFO register has been emptied after every transmission and if an error has occurred it has to send a request to the transceiver to flush the register. The risk of such an error occurring is low. Still, the radio would not be operational if that happened and that is the reason why this verification stage is important.

A state diagram describing the behavior of the receiver is shown in figure 9. On the receiving side, the microcontroller first sets the transceiver in receive mode. Similarly to transmit-mode, it then verifies that no error occurred. If an error occurred it will reset the transceiver and try again. Once the transceiver is in receive mode it will search for a certain sequence of bits that represent the beginning of the packet. Once it recognizes these bits it will output a signal on one of its digital outputs which will notify the microcontroller that a packet has been received and is ready to be read. The microcontroller will send a command to the transceiver that it is ready to extract the data. The transceiver will then proceed to send the payload of the received data to the microcontroller. The transceiver will also add 2 additional bytes at the end of the payload indicating if there are any bit errors and the signal

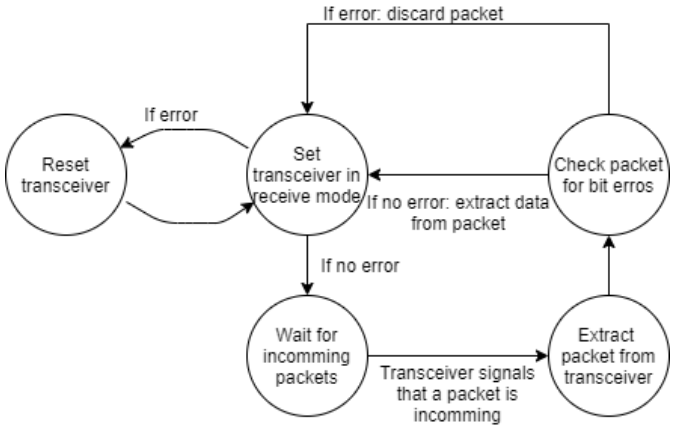


Fig. 9. State diagram of microcontroller that is operating the radio transceiver on the receiving side

strength of the packet. The microcontroller can then use these 2 bytes to verify that the data is correct and that the radio link is stable. If there are bit errors in the data it will be discarded otherwise it will be stored in a buffer so it can be sent to the user via the CAN-bus on the receiving end. Once the transceiver has received a packet it will automatically enter an idle state. Therefore the microcontroller has to manually request the transceiver to enter receive mode again as it did during the beginning of this sequence.

V. Result and analysis

The finished telemetry system on the PCB is presented in figure 10. A push-button was also mounted on the two PCBs for easier control of the radio on the fly. The system is working as intended with both communication between the microcontroller and the CAN interface, as well as RF communication between the two boards working flawlessly.

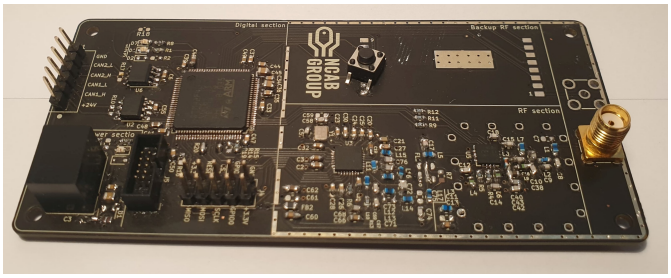


Fig. 10. One of the two final assembled PCBs

To evaluate the achievable range of the system a test was performed in Ladugårdsgärdet, Stockholm. The test was performed under good conditions with little interference from other transmitting sources, little object that could cause reflections and a clear line of sight for the duration of the test. Figure 11 shows a map of the testing environment.

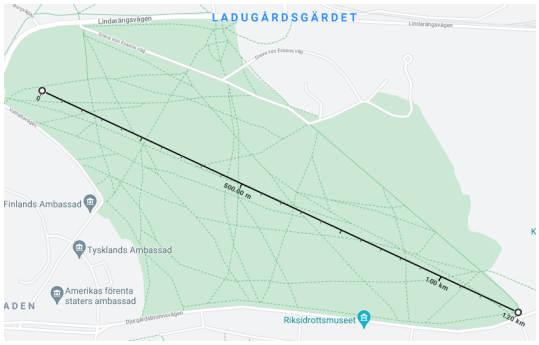


Fig. 11. The location of the range test.

The test setup is shown in figure 12. The receiving end of the setup consists of a laptop connected to the PCB via a CC-debugger from Texas Instrument. This allows for a greater overview of the received packets where packet error rate, bit error rate and average signal strength can be viewed directly. It also allows for adjusting the number of packets to be received before the average signal strength is calculated. The transmitting end consists of one of the PCBs connected to a power bank. The microcontroller on the transmitter PCB has custom software that makes it possible to swap between different data rates (50 kbit/s, 100 kbit/s and 150 kbit/s) with the push of the button soldered on the PCB. The antenna is held in a vertical position as shown i figure 12. This is the orientation the antenna will have once it is mounted in the car and also the orientation best suited for optimizing signal strength as described in IV-B2.

Fig. 12. The setup used for range tests.

The longest tested range was 1.2 km, mostly limited by the maximum line of sight achievable in the middle of Stockholm. The test started at a distance of 100m and then signal strength on the receiving end was measured every 100m up until the distance of 1.2 km. The signal strength was measured by averaging the RSSI from 1000 data packets with a payload of 25 bytes. A plot of the result is presented in figure 13. During the test, data rates of 50, 100 and 150 kbit/s were tested. They all worked without any bit errors for all 1.2 km, exceeding both the goals for date rate and range set for the project.

The output power of the telemetry system on the transmitting side was measured to 23 dBm. This was done by using an

RF power detector [12] along with a 20dB attenuation. The difference in the measured output power and desired output power of 27 dBm is 4 dB. This means that only 40% of the target power is being transmitted. It is hard to narrow down the exact reason for this difference without proper RF testing equipment and more time. Some potential reasons could be non-perfect matching networks, bad component layout, bad soldering, or an insufficient power supply.

A crude sensitivity test was performed at a data rate of 50 kbit/s. This showed that signals below -103 dBm arrived with bit errors at the receiving end. However, the radio was still able to receive some of the packets without errors. Since the RSSI is not an exact indication of the signal strength, and should only be used when talking in relative terms, this result is not expected to match perfectly with the datasheet. It does however serve a purpose in getting a clearer picture of what range to expect with the system.

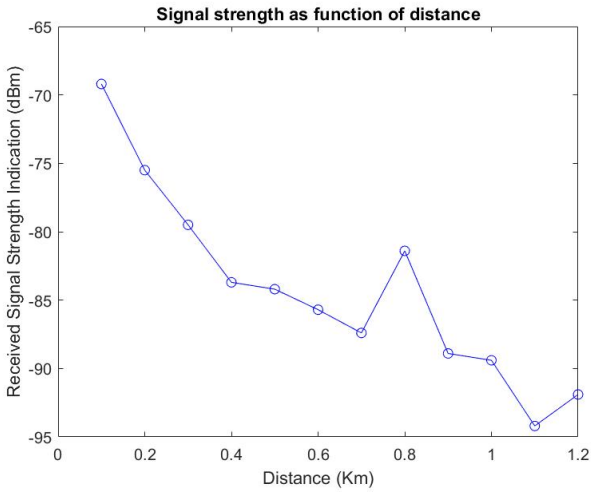


Fig. 13. The measured result from the range test i Ladugårdsgärdet

The graph shows that the desired range of 1 km was achieved with good margins. This confirms the early range estimations made using the 2-Ray ground model pretty well, even with the sub-par output power of 23 dBm. There are still a few dBm left before the signal strength reaches the approximated sensitivity of -103 dBm (RSSI) even at a range of 1.2km. Unfortunately, the test environment did not let us go any further and test even greater distances without compromising the line of sight between the radios. The graph follows a downward trend as the distance increases as expected. However, there are two exceptions to this trend at 0.8 km and 1.2 km. This can be caused by reflection from nearby objects or the ground. Another explanation could be that the height difference between the radios changed with distance due to uneven ground.

VI. Conclusion

The telemetry system exceeded all the requirements that were described in the introduction. In terms of data rate, the goal was exceeded by a factor of three, reaching data rates of 150 kbit/s. In terms of range, the distance was exceeded by 200 meters, reaching 1.2 km during a range test. This is the first telemetry system that has been implemented in the Formula Student car, which limits what can be done with the received data for now. However, it has been discovered that higher data rates than previously expected are possible which could open up for further possibilities such as low-resolution video streaming, bidirectional communication and more advanced communication protocols between the two radios. With this system now in place, more advanced testing procedures can be developed to better test and deploy new systems on the car. In a possible next revision of the system, several improvements can be made both in software and hardware. The lacking output power should be investigated further to maximize range. Additionally, steps can be taken to improve the user-friendliness of the system. For example by adding more push buttons to reconfigure the software, or a USB interface to reconfigure the system via a terminal window on a computer.

Acknowledgment

The authors would like to thank Mark Smith and Carl-Mikael Zetterling for agreeing to supervise this project. Thanks also go out to Anita Kullen for opening up the possibility of cooperation between KTH Formula Student and KTH for course work.

References

[1] M. Anand, “Short range telemetry communication for autonomous drone navigation,” in *2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, Dec, 2020, pp. 131–135.
[2] V. Torre, “Long-range body-to-body lora link at 868 mhz,” in *2019 13th European Conference on Antennas and Propagation (EuCAP)*, Jun, 2019, pp. 1–5.
[3] P. PTS and T. Authority. (2020, Dec.) Pts svenska frekvensplan. Paragraph 120. [Online]. Available: https://etjanster.pts.se/radio/undantag/foreskrifter_undantag_20202.pdf
[4] W. Electronics. (2021, Apr.) Wirl-acce 868 mhz antenna. [Online]. Available: https://www.we-online.de/katalog/datasheet/2600130081.pdf
[5] T. instruments. (2013,Jul.) Cc1200 low-power, high-performance rf transceiver. [Online]. Available: https://www.ti.com/lit/ds/symlink/cc1200.pdf
[6] E. O. Hammerstad, “Equations for microstrip circuit design,” in *1975 5th European Microwave Conference*, Sep, 1975, pp. 268–272.
[7] T. instruments. (2009, Nov.) 850 – 950 mhz rf front end. [Online]. Available: https://www.ti.com/lit/ds/symlink/cc1190.pdf
[8] T. Instruments. (2017, Sep.) Rf range estimator. [Online]. Available: https://www.ti.com/tool/RF-RANGE-ESTIMATOR
[9] T. instruments. (2013, Sep.) Cc1120 + cc1190 868-mhz reference design schematic. [Online]. Available: https://www.ti.com/tool/CC1120-CC1190EM868RD
[10] STMicroelectornics. (2021, Apr.) Stm32cubeide tool. [Online]. Available: https://www.st.com/en/development-tools/stm32cubeide.html
[11] T. Instruments. (2011, Aug.) Smartrf studio. [Online]. Available: https://www.ti.com/tool/SMARTRFTM-STUDIO
[12] C. Microwave. (2013, Jan.) Cpdetls-4000 rf power detector. [Online]. Available: https://docs.rs-online.com/6eb3/0900766b812655d9.pdf