

Kandidatexjobb i elektroteknik 2022

Kungliga Tekniska högskolan, Stockholm



PREFACE

This book contains all bachelor thesis reports from the electrical engineering programme, EECS, KTH Royal Institute of Technology, Stockholm in spring 2022. Typically, the projects have been done in 2-man groups, in only seven cases, the project was done by one student alone. This year, 65 electrical engineering students, 26 physics students, 13 vehicle engineering students, 12 energy & environment students, one computer science student and one exchange student from Spain participated successfully in the course.

The EF112X bachelor thesis course for electrical engineering students at KTH is given each spring, lasts four months, and is worth 15 ECTS credits. In addition to the project work, the course contains a series of seminars, workshops and a computer lab about the reference handling system BibTeX. The course ends with a common presentation day where all students present their project results to their supervisors and fellow students. 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 projects orally and in written form. In addition, the students reflect about the impact of the bachelor projects on society and environment and the students' future responsibilities as engineers.

The reports presented in this book are sorted into 15 different contexts. Each context starts with an introduction text, followed by the individual project reports. The introduction texts have been written by all students within the corresponding context together and consist of a popular description, a summary of the project results and a reflection about the context's importance for a sustainable society. The contexts of this year's reports are:

CONTEXT A: Automated car following and platooning
CONTEXT B: Autonomous Robotic Systems
CONTEXT C: Learning in Dynamical Systems
CONTEXT D: Embedded Systems and Motor Drives for Electric Transportation
CONTEXT E: Semiconductors for Embedded Systems
CONTEXT F: Power System Control
CONTEXT G: Power System Planning and Electricity Markets
CONTEXT H: Design and Testing of Novel Microwave/Antenna technologies
CONTEXT J: Fusion – the Sun's Energy Source on Earth
CONTEXT K: Observations in Space Physics
CONTEXT L: Observation Platforms and Instrumentation for Space Physics
CONTEXT M: Artificial Intelligence for the Internet of Things
CONTEXT N: Information Engineering: Big Data & AI
CONTEXT O: Artificial Intelligence
CONTEXT P: Big Graphs of Software Packages

In this course, the supervisors propose possible bachelor projects in advance. The students select then from a list of predefined projects, each of which has its own identification number. The project identification numbers appear in the table of content and 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., C1a and C1b). Other numbers do not appear, as those projects have not been selected 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.

It was a real joy to have all students back again at KTH Campus in March 2022 when the worst part of the corona pandemic was over. The students and supervisors did everything to ensure that the project work was successfully completed, and to make the common presentation day at KTH Campus in May a success. A few project groups got outstanding results. For example, the results of project O2b (A. Janshagen and O. Mattsson) were presented at an international workshop, another project is on the way to be published as part of a scientific paper (O. Allen and E. Skog, project O1a).

My special thanks go to the administrator Kristin Linngård for her huge administrative help, and the excellent teachers in this course which include Joakim Lilliesköld (responsible for the work plan), Martin Lindberg (organizer of the computer lab), and Anna Herland (seminars about source critics and opposition and correction of all report reviews). I myself was responsible for the seminars about written and oral communication, the intro texts, the presentation day and the organization of this course.

Anita Kullen (course responsible for EF112X)
Stockholm, September 15, 2022

TABLE OF CONTENTS

CONTEXT A: Automated Car Following and Platooning	7
A1. Platoon Coordination of Electric Trucks at a Charging Station	11
A2. Truck Platoon Coordination in a Large-Scale Transportation System	19
A3. Adaptive Cruise Control and Platooning With Tire Slip Awareness	25
CONTEXT B: Autonomous Robotic Systems	33
B1. Motion Planning for Aggressive Flights of an Unmanned Aerial Vehicle	37
B2. Collaborative Control of Autonomous Ground Vehicles	45
B3. Controlling Autonomous Baker Robot Using Signal Temporal Logic and Control Barrier Functions	55
CONTEXT C – PART I: Learning in Dynamical Systems	65
C1a. Comparison of Indirect Inference and the Two Stage Approach	69
C1b. Predictions of Electricity Prices in Different Time Periods With Lasso	75
C4a. Short Term Stock Price Prediction Using Machine Learning	83
C4b. Deep Learning Methods for Recovering Trading Strategies	91
C5. Estimating Believed Knowledge of Portfolio Agents Using Inverse Optimization	105
CONTEXT C – PART II: Learning in Dynamical Systems	115
C2a. Playing Atari Breakout Using Deep Reinforcement Learning	119
C2b. Deep Reinforcement Learning for Card Games	127
C3. Scalable Deep Reinforcement Learning for a Multi-Agent Warehouse System	135
CONTEXT D: Embedded Systems and Motor Drives for Electric Transportation &	
CONTEXT E: Semiconductors for Embedded Systems	145
D1. Battery Management System Software for a High Voltage Battery Pack	149
D2. Prototype of a Charge Controller for a Formula Student Electric Vehicle	163
D3. A General Purpose Analog Circuit to Accumulate Data From Resistive Sensors	175
D5. Design, Analysis and Implementation of a Drive System for Delsbo Electric Light Rail Vehicle	187
E2. Minnestekniker bortom halvledare för inbyggda system	195
CONTEXT F: Power System Control	211
F1. Supporting Frequency Stability With Batteries in Low Inertia Power Systems	215
F2a. Small Signal Stability of Power Systems With Increased Converter Based Power Production	225
F2b. Assessing the Impact of High Grid Penetration of Renewable Energy on Power System Stability	235
F3. Design of a Future Residential DC Microgrid	243
CONTEXT G: Power System Planning and Electricity Markets	251
G1. Voltage Stability and Reactive Power - Introduction of Intermittent Renewable Energy Sources in a Power system	255
G2a. Capacity Market in US	269
G2b. Capacity Market Design and Theory	279
G3. Modeling of Hydro-Power in Spine - Optimizing Electricity Production With a Piece-Wise Linear Dependency	291
CONTEXT H: Design and Testing of Novel Microwave/Antenna Technologies	299
H1. 3D Printed Modulated Geodesic Lens Antenna with Even Coverage in the Far-Field	303
H2. A Comparison Between Applied Square and Ring CSRR on SIW Using the HOM Method	311
H5. Design of a Leaky-wave Antenna Based on Goubau Line for Imaging Applications	319
H6. Simulation of Microwave Heating of Healthy and Cancerous Human Tissue With Gold Nanoparticles	327

CONTEXT J: Fusion – the Sun’s Energy Source on Earth	337
J1. Accelerator-Based Analysis of Rough Wall Materials From Fusion Devices	341
J3. Comparison of RF Heating in ASDEX Upgrade and ITER	353
CONTEXT K: Observations in Space Physics	367
K1. Using Jupiter’s Moon Io as a Plasma Probe	371
K2. Modelling of the Bow Shock and Magnetopause of Jupiter Using In-situ Juno Data	379
K4. Using Satellite Data to Calculate Entropy of Electrons at Collisionless Shocks	387
CONTEXT L: Observation Platforms and Instrumentation for Space Physics	395
L1. Characterisation of Satellite Onboard Magnetometer for MIST	399
L2. Wave Propagation Experiment on FPGA with Miniaturized Payload for Sounding Rocket Application	409
L4a. Building A Fixed Wing Autonomous UAV	419
L4b. Obtaining Pitch Control for Unmanned Aerial Vehicle Through System Identification	433
L5. Electric Propulsion for a High Altitude Unmanned Aerial Vehicle	445
CONTEXT M: Artificial Intelligence for the Internet of Things	457
M1. Building and Training a Fully Connected Deep Neural Network From Scratch	461
M2. Water Contamination Detection With Binary Classification Using Artificial Neural Networks	465
M4. Online Sample Selection for Resource Constrained Networked Systems	477
M5. Cybersecurity Evaluation of an IP Camera	493
M6. Integrating the Meta Attack Language in the Cybersecurity Ecosystem: Creating new Security Tools Using Attack Simulation Results	505
CONTEXT N – PART I: Information Engineering: Big Data & AI	513
N1a. Human Activity Recognition and Step Counter Using Smartphone Sensor Data	517
N1b. Step Counter and Activity Recognition Using Smartphone IMUs	527
N2a. Explaining Mortality Prediction With Logistic Regression	537
N2b. Mortality Prediction in Intensive Care Units by Utilizing the MIMIC-IV Clinical Database	547
N3. Investigation of Information-Theoretic Bounds on Generalization Error	557
N4. Experiments of Federated Learning on Raspberry Pi Boards	565
CONTEXT N – PART II: Information Engineering: Big Data & AI	575
N5a. Comparison of Discriminative and Generative Image Classifiers	579
N5b. The Impact of Noise on Generative and Discriminative Image Classifiers	591
N6a. Neonatal Sepsis Detection Using Decision Tree Ensemble Methods: Random Forest and XGBoost	601
N6b. Neonatal Sepsis Detection With Random Forest Classification for Heavily Imbalanced Data	615
N7. Robustness of Image Classification Using CNNs in Adverse Conditions	621
N8. Pre-analysis of Nanopore Data for DNA Base Calling	629
CONTEXT O: Artificial Intelligence	637
O1a. Grid-based Pursuit Evasion Games of Imperfect Information: Theory and Higher Order Knowledge-based Strategies	641
O1b. Strategy Synthesis for Multi-agent Games of Imperfect Information With Partially Given Strategies	653
O2a. Playing the Fox Game With Tree Search: MCTS vs. Alpha-Beta	663
O2b. Monte-Carlo Tree Search for Fox Game	673
CONTEXT P: Big Graphs of Software Packages	681
P1. The State of Software Diversity in the Software Supply Chain of Ethereum Clients	683

CONTEXT A

AUTOMATED CAR FOLLOWING AND PLATOONING

POPULAR DESCRIPTION

Save the world with platooning

The world is facing environmental destruction. The toxic emissions from the transport sector are polluting the planet but reducing them is difficult and costly. Platooning, where trucks drive as a unit with small distances in between, might be the solution. By taking the driver out of the equation and making the vehicles think for themselves, the distances between cars can be reduced well beyond what would be safe with human drivers. This could reduce the greenhouse gas emissions by up to 20%.

With today's increasing fuel prices transportation is becoming more and more expensive. To counter this the transportation business needs to evolve. A new transportation technology where vehicles are connected and drive closely together can save both money and the environment. This new technology is called platooning.

Most people have stood at the train station and felt the aerodynamic drag after a train has passed. This phenomenon is utilized when platooning. The drag from the foremost vehicle eases the drive for the following vehicles. An easier drive leads to a decreased energy consumption which in turn leads to a reduced fuel usage. This lowers both the fuel costs and carbon dioxide emissions.

Have you ever been stuck in traffic and wondered why the cars in front aren't moving? Very likely they are only the cumulative delay of all the drivers reacting to the car in front of them, resulting in the queue moving very slowly. By having the cars communicate in a platoon this time-costly problem could be completely eliminated. As soon as the leading vehicle begins to move the reaction of the trailing vehicles would be instantaneous and the whole queue would immediately begin moving.

Platooning has the potential of making a big economic and environmental impact on our society, as well as changing how our roads are going to function. Platooning being commercialized and put on our roads will lead to smoother traffic, a greater economic benefit and a healthier planet for all.

SUMMARY OF PROJECT RESULTS

Platooning is a concept where a set of vehicles drive in a formation with small inter-vehicular distances. Usually, the vehicles in a platoon are automated and connected to improve driving characteristics. When driving in a platoon, the trailing vehicles have a decreased air resistance which leads to a lower energy consumption. Another benefit of platooning is in principle a safer transportation system due to the lower reaction times and the elimination of the human factor. This since the driver does not need to be in control of the vehicle while in the platoon, however they can intervene manually if necessary.

Platooning is a well-researched area, however the implementation is still limited. The project groups aimed to find ways to advance platooning. To do this, project groups A1 and A2 studied ways to coordinate platoons based on maximum reward. Group A1 investigated the benefits of platooning with electric trucks while group A2 used a distributed framework to find the optimal platooning of a group of vehicles. Group A3 focused more on vehicle dynamics and cruise control with added complications such as the slip of the tire and the latency in communication.

Project group A1 investigated different strategies to coordinate electric vehicles in platoons and the potential reward of platooning with electric trucks. A model where a set of trucks with identical paths from hub A to hub B was used in the project. It was assumed that all the trucks used in the model belonged to the same fleet so that the reward could be split evenly between the trucks. Two methods to coordinate the trucks into platoons were created, where one was a time efficient way of creating platoons and one which optimized the reward. The two strategies were then used to compare the reward of platooning with electric vehicles with the platooning reward when using trucks with internal combustion engines. To analyze the trends of the result, Monte Carlo simulations in Matlab were done.

Expanding the project in future research of platooning with electric trucks could include exploring more complex models where the trucks use different routes and coordinating them to receive the biggest reward. Another way to further the project is to use multiple fleets and analyze the best way to distribute the platooning reward between the different fleets.

In project A2 a new method was proposed by the project group, which was a distributed framework where each vehicle has its own utility to optimize. A simulation of hundreds of vehicles and their respective routes were randomly created for this project. The vehicles' routes and the hubs they consisted of as well as the arrival and departure times from each separate hub for the vehicles were given. The purpose of this project was to write a program that solved the optimal waiting schedule for each individual truck so that each truck's platooning benefit was maximized. The algorithm produced information for each vehicle such as at which hub (alongside their route) the vehicle should wait, how long it should wait to join a platoon, and whether the platooning benefit would be greater than the waiting cost. According to the result, the algorithm worked as intended and the vehicles were getting platooned with a great amount of rewards.

Furthermore, this project was aiming to find the optimal platooning solution in a large-scale system and where each truck could have its own utility. By using this method, all the vehicles in the system will be getting information about at which hub they should wait and for how long they should wait, in order to maximize their own utility. Further research in this field could be to implement EU driving rules for trucks in the system, which is the constraint that a truck driver is not allowed to travel more than 4,5 hours straight without resting.

In the project of group A3 a cooperative adaptive cruise control system, C-ACC, for autonomous platooning was developed. As opposed to a regular ACC connected vehicle which is limited to the data it can obtain using radars, sensors and cameras, a C-ACC connected vehicle can exchange otherwise unobtainable information with the entirety of the platoon like the acceleration or the tire slip. The aim of the project was for the vehicles to be able to share and use knowledge about the tire slip condition in order to create a safer and more durable platoon. The tire slip is a ratio describing the difference between the velocity at the brim of the tire and the velocity of the vehicle. For example, if the wheels lock up during braking the slip is -1. The simulated platoon was exposed to harsh road conditions and then compared with and without the tire slip information in order to analyze the usefulness of including the slip. It was found that if the trailing vehicle only had access to information about the inter-vehicular distance, velocity and acceleration of the vehicle in front, then a mild deceleration, for example, could be misinterpreted as low tire capacity. To prevent this type of unwanted confusion a control law was made, utilizing the slip information, which could better communicate what was happening to the platoon at a given time. The finished C-ACC achieved satisfactory results and proved more able at dealing with changes in the road friction.

Further projects linked to C-ACC and platooning with tire slip information could include the lateral as well as the longitudinal vehicle dynamics to produce real road applicable simulations. The string stability of the platoon could also be evaluated in further detail to authenticate the results. In summary, the result of the projects within this context could help to improve platooning in general and help it to reach our roads in the future.

IMPACT ON SOCIETY AND ENVIRONMENT

The biggest and the most well-known advantage with platooning and autonomous driving is reduced energy consumption. When the vehicles are driving in a platoon with a small distance, the air resistance on the following vehicles is significantly

reduced which leads to a reduced energy consumption and cost. Secondly, a driver's workload can be reduced by joining a platoon, and combined with autonomous driving, the driver will have an easier job and the cost for employers will possibly be decreased because of this. The reduced workload can also lead to a reduction of available jobs for drivers. While it can be assumed that the opportunity for a driver to get a job will be decreased, the technology will at the same time likely increase the demand for programmers and engineers.

The reduced workload is due to features such as adaptive cruise control and lane assistance systems and can lead to decreased stress levels and more effective use of time when on the road for the driver. However, as long as a driver is needed but not fully utilized the inability to intervene can cause the driver to feel valueless.

Due to platooning being a very recent technology and still in its early phase, utilizing it efficiently might lead to it being quite expensive for companies, at least in the beginning of its development. The result could be that only large companies are able to afford using truck platooning and potentially increase their profits, while smaller companies will be left behind. While it's likely the cost of truck platooning might be too high in the beginning, collecting data and further research will probably reduce the cost of using it and make it affordable and sustainable for all companies, large and small alike.

Reducing energy consumption leads to lower carbon dioxide emissions which has a positive impact on the environment. To further decrease the CO₂ emissions, platooning can be combined with electric vehicles. Due to the lower cost when using platooning, other freight transport options may be replaced. Some of these shipping methods, such as trains, may have lower environmental consequences than transportation with trucks. To replace these freight transportation methods with truck platooning can have a negative impact on the environment in the long run.

One point of both concern and possibility for platooning is, of course, the security aspect. Driving heavy duty vehicles with only a few meters separating them seems very dangerous at first glance but could be modeled to increase the safety on our roads. The main obstacle is how to construct a safe enough controller to be able to cope with the various impacting factors affecting the platoon. If an operational controller is modeled the full benefits of platooning can be achieved. In today's traffic a large percentage of the accidents are due to human errors, an aspect the autonomous platoon will eliminate. Since the vehicles in the platoon will communicate with each other, less fluctuation between acceleration and braking occurs, therefore leading to smarter transportation. On the other hand, there is the security aspect of the system. We are seeing an increase in cyber warfare, and it is likely that this is just the beginning. An autonomous platoon is an easy target for hackers to inflict damage on a large scale, which is concerning.

A big ethical dilemma with autonomous vehicles is how they should act in high-risk situations. In the extreme case where a completely autonomous vehicle is forced to choose between injuring the people in the vehicle or injuring pedestrians, we believe that the vehicle has to always prioritize its own safety. The main reason behind this is that if the vehicle is programmed to act in a self-sacrificing manner few people will want to buy it and the whole industry will collapse. For all the positive aspects of autonomous driving to have an effect on society this problem, of accidents and collisions, will have to be dealt with in a different way, for example by building the infrastructure in a way to keep pedestrians away from the roads. If there had been a human driver in the car instead of an autonomous driver it is possible that, assuming they would have time to react, they would rather drive into a ditch than injure somebody else and be morally right to do so, but we believe that this loss is far outweighed by the general increase in safety caused by autonomous vehicles.

In conclusion there are a few problems that need to be considered before reaching the full potential of platooning. However, the positive aspects outdo the negative and therefore platooning is a promising concept for a more environmentally friendly transport system.

Platoon Coordination of Electric Trucks at a Charging Station

Elin Björklund and Ebba Lindstedt

Abstract—Electric trucks and platooning technology are expected to be part of the transportation system in the near future. Therefore, it is important to develop platoon coordination strategies and study the potential of platooning for when trucks are electric. In this paper, we study the platoon coordination problem at a single charging station where electric trucks can charge while they wait for other trucks to form platoons with. We assume all trucks to have identical routes after the charging station. The objective is to maximize the total reward of all trucks, including the platooning profit and cost of waiting. Moreover, the trucks have waiting time constraints to respect their mission deadlines and charging time constraints to make sure they can travel between the hub and destination without running out of battery. The energy consumption is decreased when driving as a follower truck in a platoon, which decreases the minimum charging time for the truck. We formulate the platoon coordination problem of electric trucks as a linear integer optimization problem. To evaluate the method, it was compared to a simpler coordination method. The savings from platooning with electric vehicles, using both coordination methods, were also compared to platooning with diesel trucks. The results showed that platooning with electric vehicles can save up to 10% of the driving cost and therefore have significant economic benefits. It was also shown that the method has an acceptable computational efficiency for real-time coordination.

Sammanfattning—Inom en snar framtid förväntas elektriska lastbilar och platooning vara en del av transportsystemet. Det är därför viktigt att utveckla strategier för platoonkoordinering och undersöka potentialen av platooning med elektriska lastbilar. I det här pappret studerar vi ett platoonkoordineringsproblem med en gemensam startpunkt och en gemensam slutpunkt för alla lastbilar. Startpunkten är en laddningsstation där lastbilarna kan kombinera laddning med att vänta in andra lastbilar att forma platooner med. Lastbilarna i systemet har även samma rutt mellan de två punkterna. Målet är att maximera den totala vinsten för alla lastbilar, med hänsyn till både platooningvinsten och kostnaden för att vänta. Utöver det har lastbilarna begränsad väntetid för att hålla sina deadlines. Vi behöver även ta hänsyn till lastbilarnas laddningstider då de behöver ha tillräckligt med laddning för att åka hela resan från startpunkt till slutdestination. Energikonsumtionen minskar när en lastbil åker som följare vilket minskar den minimala laddningstiden som behövs för att åka hela sträckan. Vi formulerar koordineringsproblemet med elektriska lastbilar som ett linjärt heltalsoptimeringsproblem. För att utvärdera metoden jämfördes den med en enklare koordineringsmetod. Besparingarna från platooning med elektriska lastbilar, med båda koordineringsmetoderna, jämfördes även med platooning med dieselbilar. Resultatet visade att platooning med ellastbilar kan spara upp till 10% av körkostnaderna och har därför betydande ekonomiska fördelar. Det visades också att metoden har en acceptabel beräkningseffektivitet för koordinering i realtid.

Index Terms—Platooning, Electric trucks, Truck coordination, E-platooning, Platoon matching, Integer linear programming

Supervisor: Alexander Johansson

TRITA number: TRITA-EECS-EX-2022:121

I. INTRODUCTION

The emissions of carbon dioxide gasses are harming the environment severely and threaten to destroy the world we live in. One of the greatest contributors to this is the transportation sector which contributed to 16.8 % of the world's greenhouse gas emissions in 2016 [1]. To lower the environmental impact of the transportation sector a lot of work is put into finding methods that produce less greenhouse gas. Two innovations to reduce the emissions are using electric vehicles and truck platooning.

A truck platoon is a formation where trucks drive with small inter-vehicular spacings [2]. An example is shown in Figure 1. To accomplish this, it is favorable if the vehicles are automated and connected. This is so that the trucks in the platoon can act together as one. With the connection, the foremost vehicle can signal the following vehicles when to brake or accelerate which the followers will do automatically [3]. Since the following trucks are automated and not as dependent on their driver, the drivers' workloads can be reduced. This, in turn, can also reduce the cost of the driver. Driving with small spacings reduces the air resistance for the trailing vehicles and eases the drive by utilizing the aerodynamic drag from the truck ahead. This leads to a decreased energy consumption for the followers which in turn results in less carbon dioxide emissions. The authors in [4] state that platooning can reduce the fuel consumption for the following vehicles by approximately 10%. A reduced fuel consumption also results in lower driving costs and therefore platooning has both environmental and economical benefits.

A. Background

Vehicles using an electric motor instead on an internal combustion engine does not produce any carbon dioxide emissions while driving. According to [5] the usage of electric trucks is increasing due to their sustainable nature. Since electric trucks need time to charge the overall travel time might be longer than the travel time with diesel trucks. However, the charging time could be utilized when coordinating platoons at hubs since trucks can charge while waiting for others. Therefore, it is interesting to investigate if platooning with electric vehicles is beneficial.

Platoons can be formed either while driving on the road or while waiting at hubs. To form platoons on the road, vehicles



Fig. 1. Trucks driving in the form of a platoon.

need to adjust the speed to connect to other trucks while driving. This coordination strategy was used in [6] which showed significant fuel savings. However, forming platoons en-route can lead to a poor traffic flow which in turn can cause congestion and accidents. To avoid this, hub-based coordination can be used. A hub can be any place throughout the route where a truck can stay and wait such as gas stations or rest areas. To form a platoon at a hub, the vehicles need to spend extra time waiting for other trucks to platoon with.

Hub-based platoon coordination was used in [7]–[12]. However, neither of these papers consider platoon coordination with electric vehicles. In [12], a method which created platoons based on optimized profit was used. Feasible platoons were found by utilizing a pairwise feasibility graph and the optimal ones were determined by solving an integer linear optimization problem. The platoon coordination method proposed in this paper is an extension of the method proposed in [12] in that our method captures the charging constraints of electric trucks. The charging time constraints are affected by whether trucks drive as followers or leaders since this affects the energy consumption.

In the recent work [13], platooning with electric trucks is considered and the authors concluded that the platoon savings are similar with electric vehicles and diesel trucks. They also concluded that platooning with homogeneous fleets, i.e., only electric trucks or only diesel trucks, is more effective due to the similar driving patterns. The system in [13] is based on day-ahead platoon coordination and the authors studied platooning with electric trucks as a way to increase the vehicles' driving ranges. In this paper, however, a system that can be used for real-time platoon coordination is considered. This, to study if the ability to depart earlier when going as a follower can lead to greater savings.

B. Problem formulation

In this project, hub-based platoon coordination with electric trucks is studied and compared with coordination of diesel trucks. A simplified system, shown in Figure 2, is considered where all trucks have the same route. They all arrive at the first hub at different times and they all have the same final destination. Platoons that maximize the total reward are

formed at the first hub with consideration of waiting and charging time constraints.

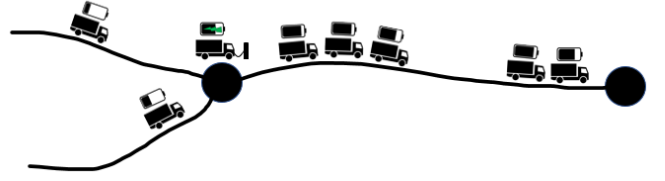


Fig. 2. Considered system with one hub where trucks can charge and platoons can form, one identical path for all trucks, and one final destination.

C. Outline

The structure of this paper is as follows. In Section II, a mathematical model of the system is presented. In Section III, the pairwise compatibility graph and the integer linear optimization problem, used for finding the reward optimizing platoons, are introduced. The simulation setup is explained in Section IV and the simulation results are presented in Section V. The result and possible future work are discussed in Section VI.

II. SYSTEM MODEL

This section introduces the model of the system by mathematically explaining basic settings and how platoons are determined. The function to calculate the total reward for a platoon is then presented which is composed of the savings in fuel and the cost of waiting at a hub.

A. Waiting and charging time constraints

The set of trucks that needs to be coordinated at the starting hub is denoted as $\mathcal{N} = \{1, 2, \dots, n\}$, where n is the number of trucks. The trucks all have the same route with distance d between the starting and finishing hub. Each truck $i \in \mathcal{N}$ arrives at the starting hub at time t_i^a . To be able to form platoons, each truck i is given a waiting time budget w_i . The electric trucks also arrive at the starting hub with a battery percentage b_i^a . To drive the distance d alone or as a leader the trucks must have the battery percentage b_i^l which is calculated with the trucks driving range, R , and the distance d . When driving as a follower the truck saves energy and therefore needs a lower battery percentage b_i^f . The earliest time when a truck can depart from the hub as a leader is given by

$$t_i^l = t_i^a + (b_i^l - b_i^a)v, \quad \forall i \in \mathcal{N}$$

and the earliest time when a truck can depart as a follower

$$t_i^f = t_i^a + (b_i^f - b_i^a)v, \quad \forall i \in \mathcal{N}$$

where v is the charging velocity. The latest time an electric truck can leave the hub is given by its waiting time budget in addition to the default departure time

$$t_i^{ld} = t_i^l + w_i, \quad \forall i \in \mathcal{N}.$$

B. Platoon feasibility and leader selection

Several electric trucks can form a platoon, p , if and only if all trucks can go as a follower before the first one must depart. This is shown in Figure 3 and given by

$$\max(t_i^f | i \in \mathcal{N}_p) \leq \min(t_j^{ld} | j \in \mathcal{N}_p) \quad (1)$$

where $\mathcal{N}_p \subseteq \mathcal{N}$ is the set of trucks in platoon p .

For a platoon to be valid there must exist a leader. A leader exists if at least one truck in the platoon can depart as a leader before the truck with the earliest departure time must leave, meaning

$$\exists i \in \mathcal{N}_p \mid t_i^l \leq \min(t_j^{ld} | j \in \mathcal{N}_p) \quad (2)$$

where \mathcal{N}_p is the set of trucks in platoon p .

The departure time for a platoon p is denoted as t_p^d and is defined as

$$t_p^d = \max(\max(t_i^f | i \in \mathcal{N}_p), \min(t_j^l | j \in \mathcal{N}_p))$$

where $\max(t_i^f | i \in \mathcal{N}_p)$ is the time when the last truck in the platoon can go as a follower and $\min(t_j^l | j \in \mathcal{N}_p)$ is the first time when a truck can go as a leader. In Figure 3 an example of time intervals for two trucks in a platoon is presented. The two trucks can depart as a platoon from the second truck's follower time, t_2^f , to the first truck's latest departure time, t_1^{ld} . The earliest departure time for the platoon in the figure is t_2^f since the first truck's leader time is earlier than this time.

Remark. We consider coordination of electric trucks in this work. However, the coordination method that we propose later can be used to coordinate diesel trucks as well by neglecting the time it takes to refuel and therefore setting $t_i^l = t_i^f = t_i^a$.

C. Platooning reward

The set of trucks in platoon p is denoted as $\mathcal{N}_p \subseteq \mathcal{N}$. For each truck in platoon p there is a cost for waiting and for each follower there is a saving. The total reward for platoon p is given by

$$r_p = S(|\mathcal{N}_p| - 1) - \sum_{i \in \mathcal{N}_p} C(t_p^d - t_i^l) \quad (3)$$

where S is the savings factor based on energy price and consumption and C is a function for the cost of waiting. This is accurate if the follower trucks have a significantly lower energy consumption than the leader truck and the trucks are homogeneous.

III. POSSIBLE PLATOONS AND PROFIT OPTIMIZATION

In this section, the concept of a pairwise compatibility graph is introduced and applied to platoon coordination. The process of finding the combination of platoons with the greatest total reward is explained where an integer linear optimization problem is utilized. An illustration of the entire process from arriving trucks to optimal platoons is also shown in Figure 4.

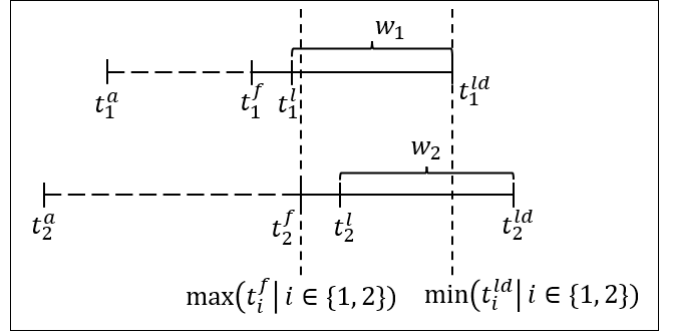


Fig. 3. Time intervals for two trucks that can form a platoon between the times t_2^f and t_1^{ld} .

A. Pairwise compatibility graph

To find feasible platoons in a set of trucks \mathcal{N} a pairwise compatibility graph (PCG) is used. In the PCG, each truck is represented by a node and an edge between two nodes exists if the two trucks can form a platoon. Two trucks, i and j can form a platoon if the condition in Equation (1), with $\mathcal{N}_p = \{i, j\}$, is fulfilled. To form a platoon, all platoon participants must be pairwise compatible with each other. An example of a pairwise compatibility graph and its feasible combinations is shown in Figure 4. From the PCG, all possible platoons from the set of trucks \mathcal{N} is found. A leader is then selected in each possible platoon by utilizing Equation (2) and then choosing the truck i with the earliest leader time t_i^l of all the possible leaders in the platoon. The set of possible platoons is denoted as \mathcal{P} .

B. Optimization problem

The platoons with the highest total reward are given by solving the integer linear optimization problem

$$\max_{x_p | p \in \mathcal{P}} \sum_{p \in \mathcal{P}} r_p x_p \quad (4a)$$

$$s.t. \quad x_p = \{0, 1\}, \quad (4b)$$

$$\sum_{p \in \mathcal{P}_i} x_p = 1, \quad (4c)$$

where x_p is the decision variable which equals one if platoon p is used and zero if not used as shown in Equation (4b). The variable r_p is the total reward for platoon p calculated with Equation (3) and $\mathcal{P}_i \subseteq \mathcal{P}$ is the set of platoons that includes truck i . The condition in Equation (4c) ensures that each truck i is a part of exactly one platoon. This optimization problem can be solved with commercial optimization solvers. The process of the platoon selection is illustrated in Figure 4.

IV. SIMULATION SETUP

This section describes the simulation setup. First, the differences when coordinating platoons with diesel trucks are explained. This is later used to evaluate the result of platooning with electric trucks. Another platoon coordination method is also introduced. The other method, which is simple and computational time efficient, is used to evaluate the reward optimization method. The chosen numeric values for all the parameters are then explained and summarized in Table I.

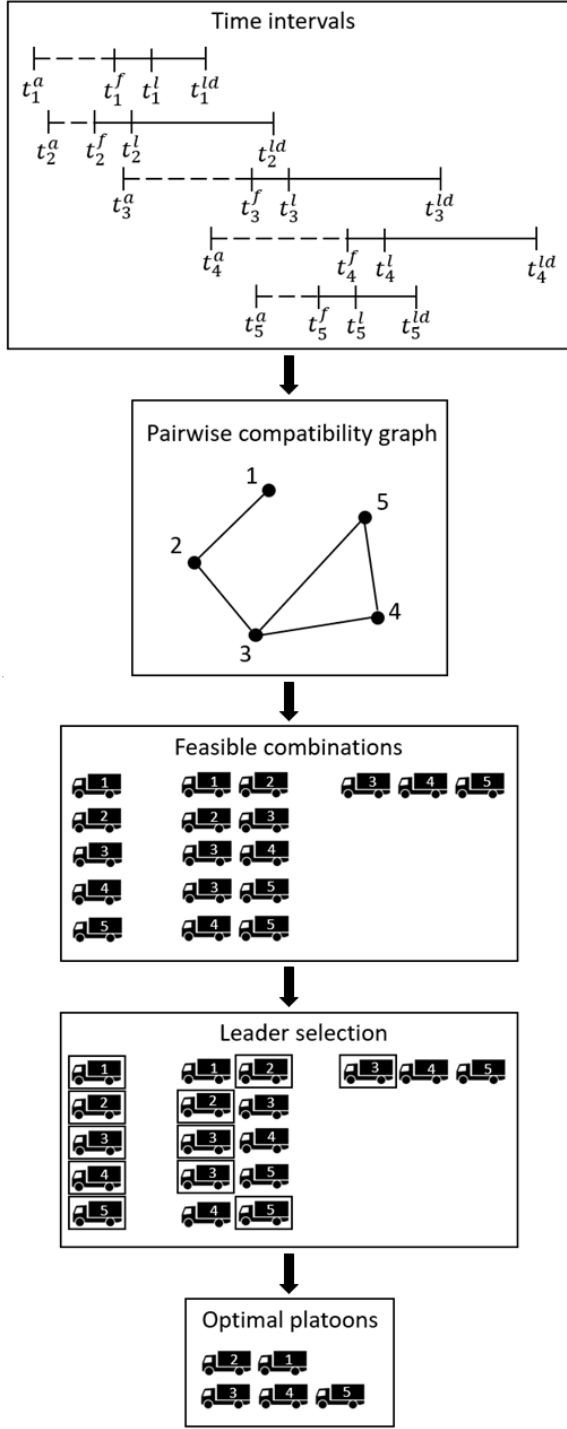


Fig. 4. The process of forming optimal platoons of a set of incoming trucks with waiting and charging time constraints. First, the pairwise compatibility graph is computed from the waiting and charging time constraints. Then, a leader is selected for each platoon. Last, the feasible platoons and selected leaders are used as an input to an optimization problem to optimize the total platooning profit.

To evaluate the performance of the proposed method, we compare it to a simplistic benchmark method where the trucks are sorted according to their earliest possible departure time. Then, we loop through the trucks and decide for one truck at a time whether it is profitable and feasible to wait for the next

arriving truck. If so, the two trucks form a platoon. Otherwise, the truck will depart on its own.

The considered system is between the cities Linköping and Stockholm in Sweden. The distance between the cities is $d = 200$ km and the number of trucks in the system is varied from $n = 1$ to $n = 50$. To avoid platoons that may cause traffic problems due to their size a maximum platoon length of $|\mathcal{N}_p| = 4$ trucks is used. Each truck is given a random arrival time in an interval of one hour and a random battery percentage between 0 and 100. Furthermore, each truck is given an allowed waiting time, w_i , between 0 and 20 minutes.

The trucks used in the simulation have an electricity consumption of $f_{electric} = 98.7/100$ kWh per km and a maximum driving range of $R = 250$ km [14]. The time it takes to charge the truck's battery from 0 to 100% is described by a function. The function is approximated to consist of two linear parts based on the battery characteristics from [15]. To charge from 0 to 80% takes 65 minutes meaning a charging velocity of $v_{0-80\%} = 80/65$ %/min. It takes 35 minutes to charge from 80 to 100% which gives a charging velocity of $v_{80-100\%} = 20/35$ %/min. Given the distance d and the range R the battery percentage a truck needs to go as a leader is $b^l = 80\%$. When reducing the fuel consumption by 10%, as a follower, the battery percentage needed for the distance is $b^f = 72\%$. The diesel trucks used in the simulation for comparison have a fuel consumption The fuel consumption for the diesel trucks used in the simulation is $f_{diesel} = 30/100$ liter per km.

The reward from platooning is determined by the followers' saved energy and time spent at hubs. The savings factor for each electric truck follower is €0.75/100 km and for each diesel truck follower €5.1/100 km. These numbers are accurate when each follower saves 10% energy, the electricity price is $f_{p_{electric}} = 0.075$ €/kWh and the diesel price is $f_{p_{diesel}} = 1.7$ €/l. The electricity price $f_{p_{electric}}$ is approximately the average electricity price during the year 2021 in Sweden [16] and to see how big price changes affect the profit another electricity price is used. This electricity price is based on the highest number in 2021 which was approximately €0.2 per kWh. The cost of waiting at hubs is considered $C_{late} = 25$ €/h. Since the electric vehicles can depart earlier when going as a follower they can save time. The profit from the saved time is approximated to $C_{early} = 12.5$ €/h which is half of the cost of waiting. All numeric values are summarized and shown in Table I. To be able to study the trends of the result Monte Carlo simulations are performed with 100 samples.

V. SIMULATION RESULTS

In this section, the results of the simulation are presented. First, the profit when platooning with electric trucks is compared to the profit when platooning with diesel trucks. We also compare our coordination method to the simplistic benchmark method explained in the previous section. The efficiency of the coordination methods is then presented.

TABLE I
NUMERIC VALUES

Name	Symbol	Value	Unit
Distance of route	d	200	km
Number of trucks	n	50	-
Maximum length of platoon	$ \mathcal{N}_p _{max}$	4	-
Maximum waiting time	w_{max}	20	min
Maximum driving range	R	250	km
Charge velocity for 0-80%	$v_{0-80\%}$	80/65	%/min
Charge velocity for 80-100%	$v_{80-100\%}$	20/35	%/min
Battery percentage leader	b^l	80	%
Battery percentage follower	b^f	72	%
Fuel consumption for electric truck	$fc_{electric}$	98.7/100	kWh/km
Fuel consumption for diesel truck	fc_{diesel}	30/100	l/km
Savings factor electric	$S_{electric}$	0.74/100	€/km
Savings factor diesel	S_{diesel}	5.1/100	€/km
Fuel price electric	$fp_{electric}$	0.075	€
Fuel price diesel	fp_{diesel}	1.7	€/l
Cost of waiting	C_{late}	25	€/h
Profit for earlier departure	C_{early}	12.5	€/h

A. Profit and platooning efficiency

The total platooning reward with the optimal solution and the benchmark solution for both electric and diesel trucks is shown in Figure 5. In Figure 5a the reward with the electricity price $fp_{electric}$ is shown and in Figure 5b the reward with the highest electricity price in Sweden 2021. The figure shows that the optimal solution leads to a larger cost reduction than the benchmark solution for both electric and diesel trucks. It can also be seen that the reward for platooning with diesel trucks is considerably larger than the reward of platooning with electric trucks for both coordination methods. This is reasonable since the fuel price for diesel trucks is higher than the electricity price. Figure 5b shows that with a higher electricity price the platooning reward increases for both methods with electric trucks. This is also reasonable because a higher fuel price will lead to a greater reward when saving 10% of the energy.

In Figure 6 the platooning reward per truck in the system is presented. One can see that the reward per truck is higher for diesel trucks and that the proposed solution achieves a greater profit than the benchmark solution. It is also shown that the reward does not increase linearly like in Figure 5 but levels out when the number of trucks grows. The maximum reward per truck is limited to the reward a truck earns when it departs as a follower. Because of the limitation, the maximum reward for each truck cannot exceed the reward when all trucks in the system are followers. Therefore, the result in Figure 6 is reasonable.

In both Figure 5 and 6 it is clear that the platooning reward is larger for diesel trucks than electric trucks. This is expected since the cost of diesel is much higher than the cost of electricity. However, the higher diesel price also leads to a higher driving cost, and therefore, this representation can be misleading. Figure 7 shows the reward in percent of the cost of driving the entire distance. Here, one can see that the reward is greater for the electric trucks than the diesel trucks.

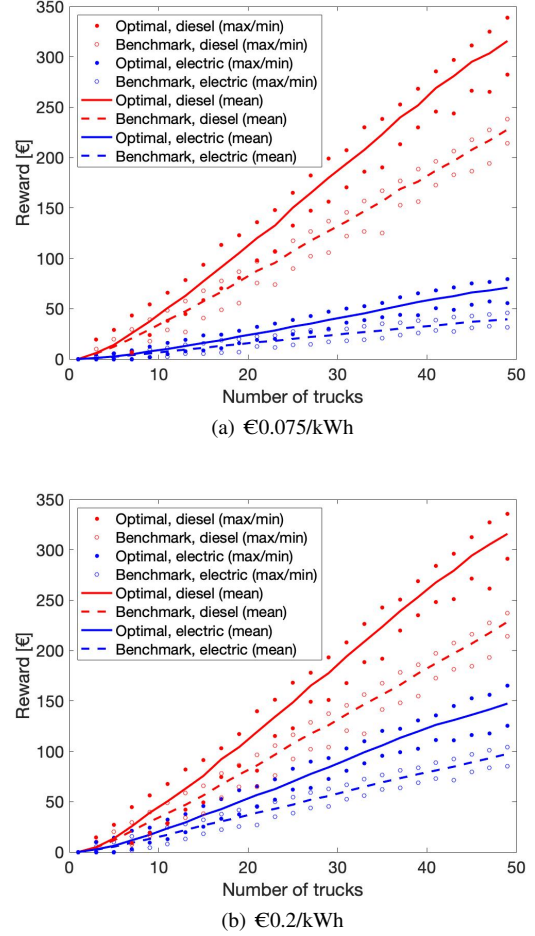


Fig. 5. The total reward from platooning with two different coordination methods, one reward optimizing and one benchmark solution, and two different fuels, electricity and diesel. In Subfigures 5a and 5b the reward is calculated with the electricity prices €0.075/kWh and €0.2/kWh respectively.

This is reasonable since the electric trucks can save money, that the diesel trucks cannot, by departing earlier. The diesel trucks must use their waiting time to form platoons while the electric trucks can depart sooner than they would have if they drove alone. This is because the following trucks in a platoon consumes less energy and therefore, the electric trucks can charge for a shorter time when departing as a follower. The used waiting time for all trucks in the system is shown in Figure 8. The figure supports the reasoning since it shows that the electric trucks save time while the diesel trucks use their waiting time and depart later.

Figure 9 shows the number of platooning trucks with both coordination methods and fuel alternatives. It can be seen that almost all of the diesel trucks in the system platoons when coordinated with the optimal solution. The diesel trucks coordinated with the benchmark solution have slightly fewer trucks participating in platoons and the electric trucks are marginally lower. This is reasonable since the electric trucks have a lower platooning reward. The diesel vehicles save more money when platooning, due to the higher fuel cost, and therefore they have more money to spend on waiting. It is also expected that the optimal solution has more platooning

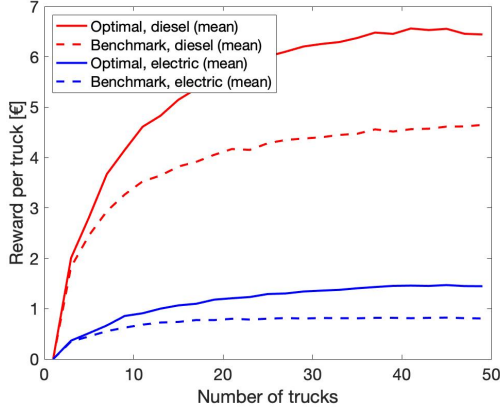


Fig. 6. The platooning reward per truck in the system with two different coordination methods, one reward optimizing and one benchmark solution, and two different fuels, electricity and diesel.

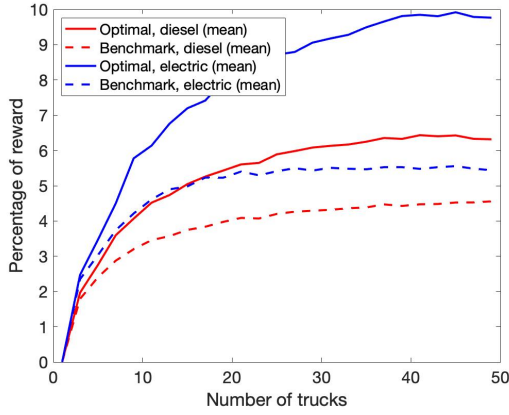


Fig. 7. The platooning reward in percent of the cost of driving the entire distance. The profit is shown with two different coordination methods, one reward optimizing and one benchmark solution, and two different fuels, electricity, and diesel.

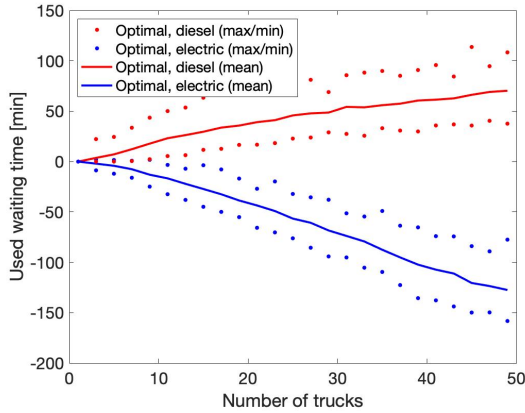


Fig. 8. The total used waiting time for all trucks in the system. The used waiting time is shown for both diesel and electric trucks.

trucks than the benchmark solution. This, because the first method finds all the possible platoons while the other method only checks for nearby pairs. For every pair that cannot form

a platoon at least one truck will depart on its own.

B. Computational efficiency

The computational efficiency of the optimal method is evaluated through a comparison with a simplistic benchmark coordination method. The computational time for the two methods with both diesel and electric trucks is presented in Figure 10. It is shown that the computational time for the optimal solution is greater than the computational time for the other method. Out of the 100 simulations, the maximum values for the different numbers of trucks do not exceed 40 seconds. However, the mean value for both the electric and the diesel trucks is below five seconds. The computational time for the benchmark method has both maximum and mean values under one second.

The number of feasible platoons is shown in Figure 11. As seen in the figure the optimal method produces a substantially higher number of found possible platoons than the benchmark solution. This is because the optimal solution uses a PCG to find all the feasible platoons while the other method only computes platoons with nearby pairs. It can also be concluded that the optimal solution with diesel trucks produces more profitable found optimal platoons than the same method with electric trucks. This may be because of the high waiting cost. Since the cost of waiting is the same for both diesel and electric trucks the difference in platooning profit makes so that diesel trucks can afford to wait for a longer time and still save money.

VI. CONCLUSIONS AND FUTURE WORK

The aim of the project was to study the benefits of platooning with electric trucks by finding a coordination method that optimized the platooning reward. To do this a simplified system was used where all trucks drive the same route and belong to the same fleet. We developed a method that first finds all possible platoons considering waiting and charging constraints and then uses the platoons that maximize the total reward.

To evaluate the result of platoon coordination with electric trucks using a reward optimizing method the result was compared to platooning with diesel vehicles. The result showed that the profit, when platooning with electric trucks, was lower than the profit with diesel trucks. This is due to the lower fuel price. The differences in fuel price also lead to different driving costs for the diesel and the electric trucks, and therefore, the reward was calculated as a percentage of the total fuel cost. When comparing the percentage reward of the two types of fuel the result was that the electric trucks achieved a higher profit. Our simulation showed that platooning with electric trucks can save almost 10% of the driving cost when we have 50 trucks in the system. This is interesting since, even though, the followers save 10% of the energy, the leaders do not save any energy. The extra savings come from the ability to depart earlier. It was shown that all 50 trucks in the system together saved approximately 125 minutes. However, there are other factors, that were not considered, that may affect the total cost and reward. For example, in a bigger system, electric trucks

will have a longer overall travel time since they must charge throughout the route. Electric vehicles and diesel vehicles also have differences in purchase price and durability.

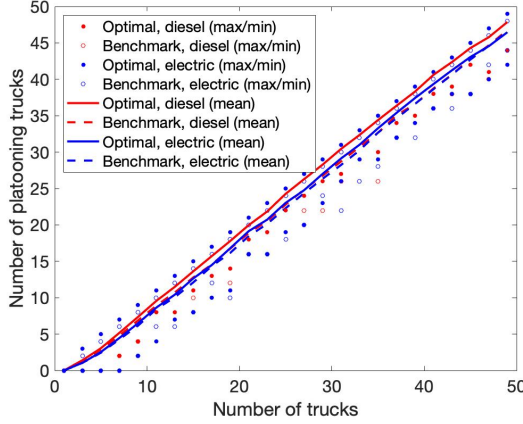


Fig. 9. The number of platooning trucks with the optimal method and the benchmark solution for both electric and diesel trucks.

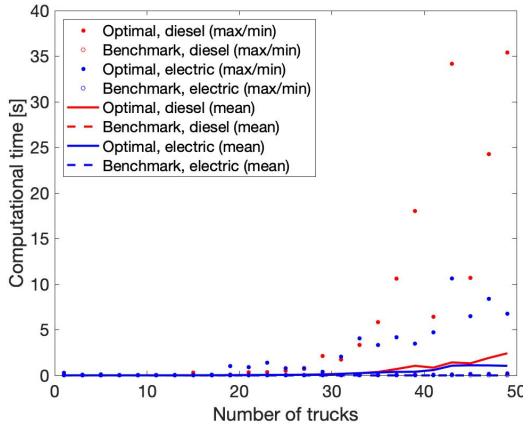


Fig. 10. Computational time for the optimal solution and the benchmark solution with both electric and diesel trucks. The computation time was computed 100 times and the minimum, maximum and mean value is shown.

The computational efficiency of the proposed coordination method was evaluated by a comparison with a simple coordination method. The computational time and the number of feasible platoons were compared, and the result showed that the other method was better in both aspects. Because of the great number of feasible platoons found in our method, it is reasonable that the computational time for that method is higher. The great number of feasible platoons is necessary to find the platoons with the highest profit, but it is demanding from a computational efficiency point of view. This is due to the many variables in the integer linear optimization problem. However, the result showed that the computational time for computing a solution for 50 trucks was, on average, less than 5 seconds which indicates that the proposed method can be used for real-time platoon coordination.

In conclusion, a profit optimizing method for platoon coordination with electric trucks has been proposed. The result from the proposed method shows that platooning with electric

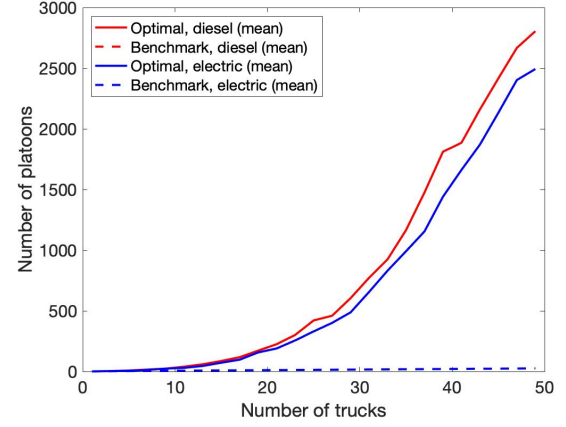


Fig. 11. Number of feasible platoons for electric and diesel trucks using both the benchmark method and the optimal method.

vehicles has great economic benefits and that the coordination method can be used for real-time coordination. There are still many developments that can be made for more realistic results however these results show that platooning with electric trucks is a promising concept with both environmental and economical benefits.

A. Future work

To expand the research in this area several improvements can be made. First, the charge velocity was approximated with two constant values. The battery capacity does not increase linearly when charging and therefore a better approximation might give a more reliable result. Secondly, the scale of the system can be increased to be more accurate to reality. This can be done by adding more hubs and trucks to the system. To expand it further the trucks may have different routes and belong to different fleets. Adding various types of electric trucks with varying ranges and consumptions can increase the complexity of the system even further.

ACKNOWLEDGMENT

The authors would like to thank their supervisor Alexander Johansson for his guidance throughout the project as well as his commitment and support.

REFERENCES

- [1] H. Ritchie, M. Roser, and P. Rosado. (2020, Aug.) CO2 and greenhouse gas emissions. [Online]. Available: <https://ourworldindata.org/co2-and-other-greenhouse-gas-emissions>
- [2] X. Chen and J. Mårtensson, "Optimization based merging coordination of connected and automated vehicles and platoons," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, Sep. 2021, pp. 2547–2553.
- [3] A. Hurtado-Beltran, H. Vakilzadian, and L. R. Rilett, "Impact of the entry time model on connected and automated vehicle (CAV) platoon formation," in *2020 IEEE International Conference on Electro Information Technology (EIT)*, Aug. 2020, pp. 655–662.
- [4] F. Browand, J. McArthur, and C. Radovich, "Fuel saving achieved in the field test of two tandem trucks," *UC Berkeley: California Partners for Advanced Transportation Technology*, Jun. 2004. [Online]. Available: <https://escholarship.org/uc/item/29v570mm>

- [5] Volvo. (2022, Feb.) Volvo lastvagnar marknadsledare inom helelektriska lastbilar i europa. [Online]. Available: <https://news.cision.com/se/ab-volvo/r/volvo-lastvagnar-marknadsledare-inom-helelektriska-lastbilar-i-europa,c3507345>
- [6] K.-Y. Liang, J. Mårtensson, and K. H. Johansson, "Heavy-duty vehicle platoon formation for fuel efficiency," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1051–1061, Nov. 2016.
- [7] W. Zhang, E. Jenelius, and X. Ma, "Freight transport platoon coordination and departure time scheduling under travel time uncertainty," *Transportation Research Part E: Logistics and Transportation Review*, vol. 98, pp. 1–23, Feb. 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1366554516304938>
- [8] R. Larsen, J. Rich, and T. K. Rasmussen, "Hub-based truck platooning: Potentials and profitability," *Transportation Research Part E: Logistics and Transportation Review*, vol. 127, pp. 249–264, Jul. 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S136655451830944X>
- [9] N. Boysen, D. Briskorn, and S. Schwerdfeger, "The identical-path truck platooning problem," *Transportation Research Part B: Methodological*, vol. 109, pp. 26–39, Mar. 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0191261517305970>
- [10] A. Johansson, E. Nekuoei, K. H. Johansson, and J. Mårtensson, "Strategic hub-based platoon coordination under uncertain travel times," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, May. 2021.
- [11] T. Bai, A. Johansson, K. H. Johansson, and J. Mårtensson, "Event-triggered distributed model predictive control for platoon coordination at hubs in a transport system," in *2021 60th IEEE Conference on Decision and Control (CDC)*, Dec. 2021, pp. 1198–1204.
- [12] A. Johansson, J. Mårtensson, X. Sun, and Y. Yin, "Real-time cross-fleet Pareto-improving truck platoon coordination," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, Sep. 2021, pp. 996–1003.
- [13] J. Scholl, N. Boysen, and A. Scholl, "E-platooning: Optimizing platoon formation for long-haul transportation with electric commercial vehicles," *European Journal of Operational Research*, Apr. 2022. [Online]. Available: <https://doi.org/10.1016/j.ejor.2022.04.013>
- [14] D. Burul and D. Algesten. (2021, Jun.) Life cycle assessment of distribution vehicles. [Online]. Available: <https://www.scania.com/content/dam/group/press-and-media/press-releases/documents/Scania-Life-cycle-assessment-of-distribution-vehicles.pdf>
- [15] Scania. (2020, Sep.) Scania launches fully electric truck with 250 km range. [Online]. Available: <https://www.scania.com/group/en/home/newsroom/press-releases/press-release-detail-page.html/3768729-scania-launches-fully-electric-truck-with-250-km-range>
- [16] Fortum. (2022, Apr.) Jämför elpriser - aktuella och historiska. [Online]. Available: <https://www.fortum.se/privat/elavtal/elpriser>

Truck Platoon Coordination in a Large-Scale Transportation System

Guanyu Lin and Robin Ganguly

Abstract—Truck platooning is a technology where trucks drive in a formation with each other with a small distance in between trucks in order to save fuel and reduce emissions. In this project, a distributed method for solving the optimal time problem for every truck in a hub-based transport system will be developed. Each truck will have its own utility function to optimize and is able to adjust its schedule independently. To create and test the method, a simulation of hundreds of trucks in a network of routes was created using the Python language. The results produced by running the simulation were positive and realistic.

Sammanfattning—Konvojkörning med lastbilar är en teknologi där lastbilar kör i en formation med varandra med små avstånd mellan lastbil för att spara på bränsle och minska utsläppen. I det här projektet kommer en distribuerande metod för att lösa det optimala tidsschemat för varje lastbil i ett navbaserat transportsystem att utvecklas. Varje lastbil kommer att ha sin egen vinstfunktion att optimera och kommer självständigt att kunna ändra sitt reseschema. För att skapa och testa metoden kördes en simulation som skrevs i Python, och som behandlade hundratals lastbilar i ett nätverk av vägar. Resultaten som simulationen producerade var positiva och realistiska.

Index Terms—Platoon coordination, model predictive control, hub-based transport system

Supervisors: Ting Bai

TRITA number: TRITA-EECS-EX-2022:122

I. INTRODUCTION

Truck platooning is a modern technological concept where two or more trucks are linked together in a formation and travels on their common route with small inter-vehicular distances between each other in order to get a decreased air drag and operational cost. The first truck in the platoon is the leader, and the others are the followers. When driving in a platoon, the followers will automatically react and adapt to the leader's movement, which require much less actions from the drivers. The fuel saving is related with the distance between trucks. According to the research [1], trucks driving in a platoon with a 4 meters gap will give a 4.8% CO_2 saving.

Nowadays, more than 6.2 million trucks are in the circulation throughout the European Union per year and they are carrying 73.1% of all freight transported over land in the EU [2]. These amounts cause up to 25% of road transport emissions in the EU alone. In reality, it is not easy to reduce CO_2 emission from vehicles by improving the fuel efficiency of new vehicles, because the efforts to improve it are slowing in recent years due to the technological barriers. This means that a new solution is needed for reducing emissions

in order to achieve the EU target of 30% CO_2 reduction for new heavy-duty vehicles by 2030 [3].

Truck platooning is a promising way to solve the above problem and to achieve the emission goals because of its high applicability and efficiency. Truck platooning does not require trucks to have exactly the same route and destination in order to form a platoon, but only a common route segment. Due to the high amount of trucks on the highways across the whole EU, it is easy to find platoon partners that shares at least one common route segment. These trucks with common routes can then form a platoon together and begin earning rewards and reducing emissions.

The motivation to this project is to find a distributed hub-based method to form truck platoons in a large-scale transport system where each truck in the system has its own utility to optimize. For example truck A is delivering medicine which is in urgent need and its deadline is slightly above the time it takes to travel its designated route, then its maximum waiting time attribute can be set in order to obey the deadline. If truck B is delivering flour and its intent is to save as much as possible, then its maximum waiting time can be adjusted in order to let it join the most beneficial platoon for itself.

A. Related works

In the past few decades, truck platooning has been researched widely. Most of those researches revolve around how trucks in a platoon can communicate with each other [4] or about the driving safety in a platoon [5]. Truck platooning technology has been greatly improved so that the safety and the efficiency in the platoon are excellent. In the latest test [5], a truck can even be driving safely with a time gap of 0.5 second from the previous truck.

However, the problem about how to form an optimal platoon in a large-scale transportation system has not been fulfilled yet. Based on existing research [6], they are solving the problem in different ways. Some of those researched are doing a non-hub-based method, which is sensor based [7] [8] and form a platoon by controlling the speed of trucks when trucks are driving. Some of them are hub-based but centralized [9] [10] [11], that with a centralized system they were assuming that all the trucks are the same and have the same attribute and utility, such as delivery deadline, speed and travel routes, etc. But this strategy is not suitable for a large-scale transportation system with hundreds of trucks. Because trucks have different priorities and tasks, the platoon method should be considered individually.

In this paper, a distributed framework for handling the large-scale platoon coordination problem is proposed based

on the model predictive control (MPC) method, where each truck allows to optimize its own platooning benefit given the information of other trucks. This work is based on our supervisor Ting Bai's earlier work [12], but is recreated in this project with new simulations to test various scenarios. Our simulation results show that, trucks are getting a higher average platooning rates and rewards in a larger system, with a lightly increased average waiting time.

B. Contribution

Our method are coordination hub-based, which means that different hubs are set on each route which divides each truck's route into segments. Taking this as a basis, trucks are able to find proper platoon partners on a common route segment while do not need to have exactly the same route with other trucks. This will greatly simplify the process and is much more suitable in a large-scale transport system because many of the trucks' routes will overlap with each other at some parts.

Conclusively, our method is distributed, which means that each truck has its own utility and attribute. The method is also suitable for a large-scale system with a large number of trucks, which is more flexible to cope with changes in the system.

II. PROBLEM FORMULATION

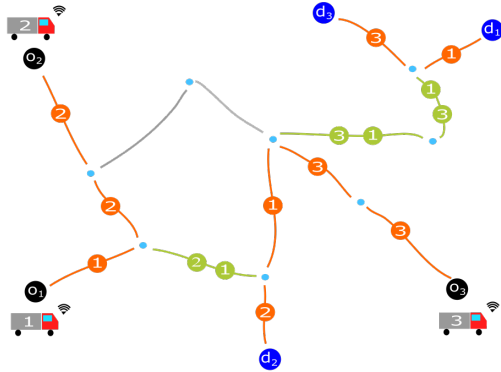


Fig. 1. A graph representing the transportation system that is considered in this project. The hubs in the system are represented with a cyan dot. Hubs that are origins and destinations for individual trucks are represented by black and blue dots, respectively. The route of each truck is denoted by segments of orange lines between hubs. If two or more trucks share a common line segment, then that segment is denoted by a green line.

For this project, the focus will be on a large-scale transportation system consisting of M trucks owned by different fleets. Every truck i , where $i \in \mathbf{M} = \{1, 2, \dots, M\}$, has a fixed route which involves N_i hubs in it, and has a set of hubs $\mathbf{H}_i = \{h_{(i,0)}, h_{(i,1)}, \dots, h_{(i,N_i-1)}\}$. The origin o_i and destination d_i of truck i correspond to the first and last hub in truck i 's route, i.e. $o_i = h_{(i,0)}$ and $d_i = h_{(i,N_i-1)}$. An arbitrary hub in truck i 's route will henceforth in this report be denoted as $h_{(i,k)}$, where $k \in [0, N_i - 2]$.

Another set that must be defined is the set of route segments that truck i 's route consists of. For any hub $h_{(i,k)}$, $e_{i(k,k+1)}$ is defined to be the directed route segment of truck i from its

k -th hub to its next $(k+1)$ -th hub. The route of truck i is then defined as the sequence of all of its directed route segments, formulated as

$$\mathbf{e}_i = \{e_{i(0,1)}, e_{i(1,2)}, \dots, e_{i(N_i-2, N_i-1)}\} \quad (1)$$

To simplify approaching the problem, the hubs along truck i 's route, the origin and destination of truck i , its depart time from the origin and the deadline truck i must adhere to, as well as the travel time between each hub along trucks i 's route are considered fixed and known. Forming platoons can only occur at hubs and therefore the problem can be formulated as such: calculate the optimal waiting schedule for every truck at each of its hubs along its route so that its platooning benefit is maximized without violating its deadline.

In order for truck i to form a platoon with another truck j at hub k , they must have the same directed route segment from hub k , i.e. $e_{i(k,k+1)} = e_{j(k',k'+1)}$, where $h_{(i,k)} = h_{(j,k')}$ and $h_{(i,k+1)} = h_{(j,k'+1)}$.

A. Model predictive control

For this project, Model Predictive Control (MPC) is used to solve this problem. The reason is because MPC can predict future behavior in a system given the system's current state and input, which is suitable in this case since each truck's predicted schedule is known.

If the distance between each hub in truck i 's route is known, one can predict from its k -th hub the arrival at its next $(k+1)$ -th hub, where $k \in [0, N_i - 2]$. The departure time of truck i from its k -th hub is defined as

$$t_{di}(k) = t_{ai}(k) + t_{wi}(k), \quad (2)$$

where $t_{ai}(k)$ is truck i 's arrival time, $t_{wi}(k)$ is its waiting time and $t_{di}(k)$ is its departure time at hub k . This can then be used to predict its arrival time at its $(k+1)$ -th hub as follows

$$t_{ai}(k+1) = t_{di}(k) + t_{li}(k), \quad (3)$$

where $t_{ai}(k+1)$ is truck i 's arrival time at its $(k+1)$ -th hub and $t_{li}(k)$ is the time it takes to travel between truck i 's k -th and $(k+1)$ -th hub. By inserting equation (2) into equation (3) you get

$$t_{ai}(k+1) = t_{ai}(k) + t_{wi}(k) + t_{li}(k). \quad (4)$$

Equation (4) describes how a truck can predict its arrival time at the next hub by using the arrival time and waiting time of the current hub and the travel time between the two hubs. With this model, a truck can predict from its origin o_i the arrival time at each hub all the way to the destination d_i .

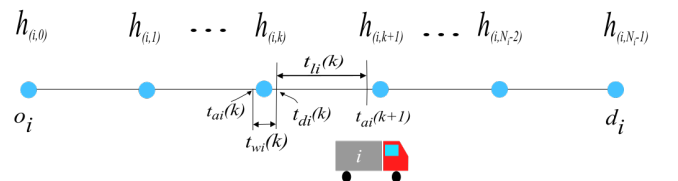


Fig. 2. Truck i 's route from its origin o_i to its destination d_i .

$$t_{ai}(N_i - 1) \leq t_i^{dd}. \quad (5)$$

Equation (5) describes the time constraint that each truck's arrival time at the destination should not be later than the truck's set deadline.

B. Predicted partners set

During the travel, every time truck i arrives at a new hub it is going to check whether it is able to form a platoon with other trucks. The first step is to check if any other trucks in the system shares the same route segment as truck i from its k -th hub; finding these trucks can be done offline for every hub in truck i 's route since every truck in the system has a fixed known route. The set of trucks that share a common route with truck i at its k -th hub is called the potential partner set and is defined as

$$\mathbf{P}_i(k) = \{j : e_{i(k,k+1)} \in \mathbf{e}_j, j \in \mathbf{M}, j \neq i\}. \quad (6)$$

After the potential partners has been obtained, the predicted partners can be calculated in real time and is defined as

$$\mathbf{T}_i(k+h|k) = \{j : j \in \mathbf{P}_i(k+h) \wedge t_{ai}(k+h|k) + t_{wi}(k+h|k) = t_{aj}(k+h) + t_{wj}(k+h)\}. \quad (7)$$

What equation (7) describes is how truck i from its k -th hub calculates its predicted platoon partners at its $(k+h)$ -th hub, where $h \in [0, N_i - 2 - k]$. Truck i 's predicted departure time from hub $(k+h)$, predicted from hub k , is $t_{ai}(k+h|k) + t_{wi}(k+h|k)$, and if this equals truck j 's departure time from hub $(k+h)$ then forming a platoon with truck j is possible. Therefore what $\mathbf{T}_i(k+h|k)$ represents is the group of trucks that truck i can form a platoon with at its $(k+h)$ -th hub calculated from its k -th hub.

C. Utility

In order to define the performance of the method and to find the optimal solution for each truck, a utility function U_i for truck i should be defined first.

According to earlier works by [13] and [14], it is assumed that followers in a platoon are driving with 10% decreased air resistance than if they are driving alone. The platoon reward function for truck i can then be defined as

$$R_i(k) = \sum_{h=0}^{N_i-2-k} t_{li}(k+h) \frac{|\mathbf{T}_i(k+h|k)|}{|\mathbf{T}_i(k+h|k)|+1} * \delta \quad (8)$$

where $t_{li}(k+h)$ is the travel time from hub $(k+h)$ to the next hub, the expression $|\mathbf{T}_i(k+h|k)|$ is the number of other trucks in the platoon truck i joins from hub $(k+h)$, and δ is the platooning benefit coefficient and describes the monetary benefit in fuel savings per follower truck per time unit. In practice, the leader can also have some fuel saving, but very small. So it is assumed that the leader gets nothing by forming a platoon while the total platooning benefit is equally shared between the leader and follower trucks so that every truck has the same willingness to join a platoon. It is assumed that all

the $|\mathbf{T}_i(k+h|k)|+1$ trucks in the platoon are getting the same equal benefits, so the reward is divided between all the trucks.

When a truck is waiting at a hub to join a platoon, its total delivery time will be extended by the waiting time. This will cause a platooning loss because of a truck driver's salary. According to the average truck driver's salary, it is assumed that

$$L_i(k) = \sum_{h=0}^{N_i-2-k} t_{wi}(k+h|k) * \epsilon \quad (9)$$

where $t_{wi}(k+h|k)$ is the predicted waiting time of truck i at its hub $(k+h)$ predicted from hub k and ϵ is the loss coefficient, which is set as the average salary per hour for a truck driver in Sweden.

Finally there is the utility function for a platoon combination for truck i at its k -th hub which is.

$$U_i(k) = \sum_{h=0}^{N_i-2-k} R_i(k+h|k) - L_i(k+h|k). \quad (10)$$

The final utility for a truck is the sum of rewards and losses along all its route segments. And $R_i(k+h|k)$, $L_i(k+h|k)$ are the predicted rewards and losses for the whole route computed at the current hub.

D. Platoon coordination method

In order to be clear on the optimization problem, the MPC model is stated here, for truck i at its k -th hub in its route. The platoon coordination problem can be formulated as follows, where the desire is to optimize the final utility by finding the optimal waiting time at each hub:

$$\begin{aligned} \max_{t_{wi}(k)} \quad & U_i(k) \\ \text{s.t.} \quad & t_{ai}(k|k) = t_i^a(k) \\ & t_{ai}(k+1|k) = t_{ai}(k|k) + t_{wi}(k|k) + t_{li}(k) \\ & t_{ai}(N_i-1|k) \leq t_i^{dd}. \end{aligned} \quad (11)$$

$$\mathbf{t}_{wi}(k) = [t_{wi}(k|k), t_{wi}(k+1|k), \dots, t_{wi}(N_i-2|k)]$$

The utility function is given by the equation (10), and the constraints are according to equation (5).

The program begins with generating a timeline according to original schedule of all the trucks. In the timeline, information such as at which minute, which trucks will arrive at which hub will be included.

When a truck i arrives at a hub in its route, except for the destination, an MPC problem will be solved in order to optimize the utility at this hub for the truck i and its whole travel schedule will be updated with the proposed waiting time. By doing this, the other trucks in the system can easier calculate their own optimized solution.

First of all it will find all the potential partners according to equation (6). Feasible partners means that the partner should have a later departure time from hub k than truck i 's arrival time t_{ai} at hub k . For each feasible platoon combination at $h_{(i,k)}$, it will be represented by a node that includes information such as the number id for all the trucks in the

platoon and how long the waiting time t_{wi} must be for truck i to form or join this platoon. Once all the partners have been added, the function will go to the next hub $k + 1$, and do the same procedure for each node until the truck arrives at its destination. Then all the feasible platoon combinations will be available.

Once all possible combinations are found, the program will do follows.

- Remove all combinations with a later arrive time at the destination than the delivery deadline.
- Calculate the final utilities with the equation (10) for all the remaining combinations and find the optimal one.
- Execute the decision to wait at current hub with the waiting time from the optimal combination.
- Update the truck's schedule on following hubs with the optimal solution for other trucks to use.

The time line will be updated after every solved MPC problem, which makes sure that the following trucks are using an updated schedule when calculating their solutions. After the last truck arrives at its destination, the program will plot the result in different diagrams for then to be analysed.

III. SIMULATION RESULT AND DISCUSSIONS

A. Simulation parameter

The simulation took place on the Swedish road network, where 84 major hubs were selected and where routes were generated for each truck based on these hubs. Hundreds of trucks were used in the simulation in order to test how our method was working in a large-scale system.

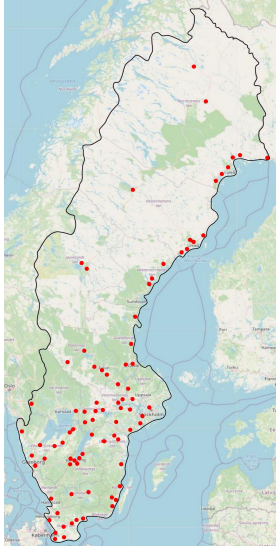


Fig. 3. 84 hubs on the Swedish road network

All the trucks were assumed to depart from their origins between 8:00 a.m. and 9:00 a.m.. The trucks were allowed to wait 30 minutes in total during their travels. Equation (10) was used for calculating the utilities for trucks with the benefit coefficient $\delta = 57.6$ SEK per hour, and the waiting cost $\epsilon = 260$ SEK per hour.

The simulation results for each single truck in a system with 100 trucks are provided in Figs. 4, 5 and 6. In order to study

how the result will be effected by the number of trucks and the maximum allowed waiting time, the system was simulated with 300, 500 and 100 trucks with a maximum waiting time of 15 minutes. The results are provided in Tables I-IV.

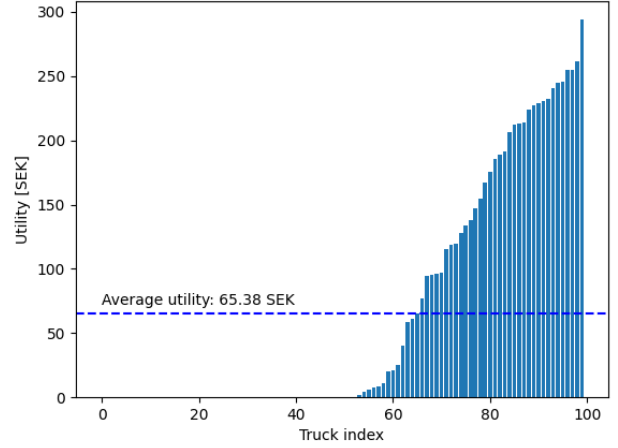


Fig. 4. Utility for each truck in 100 trucks and the average utility with max waiting time 30 mins

The result in Fig. 4 is as expected, where trucks get an average utility of 65.38 SEK, the highest reward for a single truck is almost at 300 SEK and around 50 of 100 trucks get rewards.

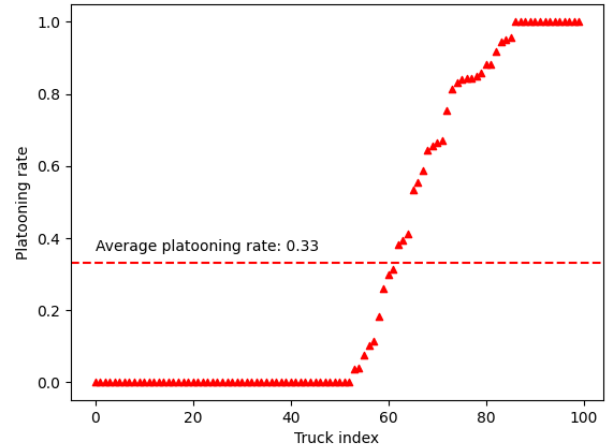


Fig. 5. Platooning rate for each truck in 100 trucks and the average platooning rate with max waiting time 30 mins

Here the term "platooning rate" is introduced in order to evaluate the coordination result. Platooning rate for a truck is defined as:

$$Pr = \frac{\text{its total travel time in a platoon}}{\text{its total route time}}, 0 \leq Pr \leq 1. \quad (12)$$

As can be seen in Fig. 5, there are around 17 trucks with a platooning rate at 1, which means they are driving their entire routes while being in a platoon. There is an average platooning rate of 0.33, and around 50 of 100 trucks have a non-zero platooning rate.

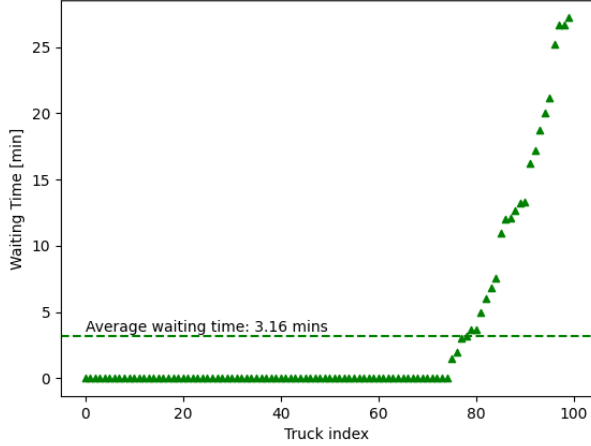


Fig. 6. Waiting time for each truck in 100 trucks and the average waiting time with max waiting time 30 mins

According to the Fig. 6, almost 67% of the trucks do not need to wait and only less than 5 trucks waited for more than 25 minutes. The average waiting time across all the trucks is 3.16 minutes, which is around 10% of the maximum waiting time for all the trucks.

From the test with a total of 100 trucks and with the maximum waiting time of 30 minutes it is clear that the method was working efficiently, where around half of the trucks were getting a platooning benefit and they all had an average reward of 65.38 SEK with only an average waiting time of 3.16 minutes.

TABLE I
AVERAGE PLATOON REWARD TEST

Nr. trucks	Maximal waiting time	Average platooning reward
100	15 mins	42.41 SEK
100	30 mins	65.38 SEK
300	30 mins	85.65 SEK
500	30 mins	92.84 SEK

Table. I shows that the average platooning reward got a significant improvement when the number of trucks increased, and when the maximum waiting time got decreased to half of 30 mins, the average platooning reward is decreased from 65.38 SEK to 42.41 SEK, around to 64%. This indicates that the average platooning reward and the amount of trucks in the system are a positive correlation. Platooning rewards will be decreasing if the maximum waiting time becomes less.

TABLE II
AVERAGE PLATOONING RATE

Nr. trucks	Maximal waiting time	Average platooning rate
100	15 mins	0.20
100	30 mins	0.33
300	30 mins	0.46
500	30 mins	0.50

The platooning rate is related to platooning reward. The platooning rate also had a positive correlation with the amount of trucks and with the waiting time. If a higher platooning

rate is desired, the number of trucks can be increased or be allowed to wait for a longer time. The result is reasonable because if there are more trucks on the route then there will be more opportunities to form platoons for all the trucks, and the platooning reward and platooning rate will both be higher.

TABLE III
AVERAGE WAITING TIME

Nr. trucks	Maximal waiting time	Average waiting time
100	15 mins	1.33 mins
100	30 mins	3.16 mins
300	30 mins	4.70 mins
500	30 mins	5.00 mins

According to the result shown in Table. III, the average waiting time of all trucks in the system wasn't affected as much as the average platooning rate and average platooning reward by trucks' amount. It was more sensitive to the maximum waiting time attribute. When the maximum waiting time changed from 30 minutes to 15 minutes, the average waiting time decreased from 3.16 to 1.33 minutes, it was almost 42% of itself. But the average waiting time only increased from 3.16 minutes to 4.70 minutes, which increased to 145% of itself, when the number of trucks increased for three times, from 100 trucks to 300 trucks.

Conclusively, the average platooning rate and the average reward are correlated with both the amount of trucks and the maximum waiting time. When the number of trucks increases or the maximum waiting time increases, there is a significant increase in platooning reward and platooning rate. The average waiting time depends more on the maximum waiting time attribute, but also trucks' amount. When the maximum waiting time decreases, the average waiting time will follow and when there are more trucks in the system, the average waiting time will slightly increase.

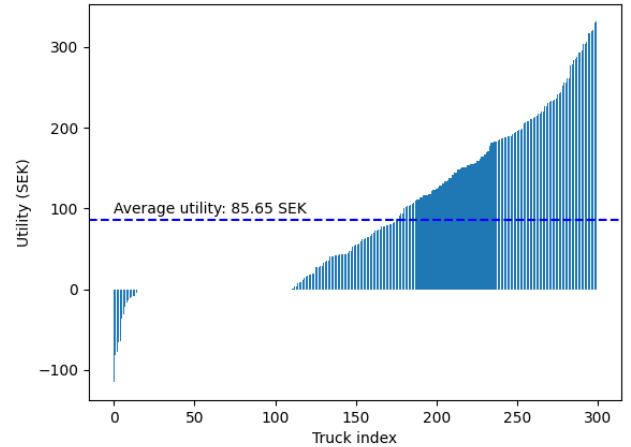


Fig. 7. Some trucks got negative reward in a transportation system with 300 trucks

It was found that some trucks were getting negative reward in a larger system(simulations with 300 and 500 trucks). The major reason is the waiting decision of a truck is only based on

the predicted schedules of other trucks. While the other related trucks are free to change their decisions and predictions. For example if truck A made a decision to wait at a hub in order to form platoon with truck B. But truck B didn't choose to form platoon with truck A because they had a better partner truck C to form a platoon with.

The desire is for every truck to optimize its own utility, so when a truck finds a better combination that the truck can be earning more utilities by joining that platoon, the truck will choose that combination. That's the reason that the predicted schedule used by other trucks might be wrong in reality. A way to solve this problem would be that truck A, before waiting for truck B, can always send a request to truck B, and only wait for truck B if truck B accepts the request, but this will effect the efficiency and slow down the program in a large-scale system.

TABLE IV
TRUCKS WITH NEGATIVE REWARDS

Nr.trucks	Average negative reward	Nr. trucks with negative rewards	Ratio
100	0	0	0
300	-35.45 SEK	16	0.053
500	-29.06 SEK	17	0.034

Around 5.3% of trucks were getting negative utilities in the simulation with 300 trucks, and 3.4% in the simulation with 500 trucks. The average negative reward is similar in both simulations. The result shows that the negative reward issue is not directly related with the number of trucks in the system, and this issue can be ignored comparing to the system gain.

IV. CONCLUSION AND FUTURE WORKS

In this paper, an optimal platoon coordination problem in a large-scale transportation system has been solved by a distributed MPC method. Hundreds of trucks in the system can form platoons effectively and get their optimal utilities. The trucks at waiting hubs can calculate and predict their rewards and loses by our method, and then make their decisions to optimize their utilities. Every truck has its own utility and is able to make its own decision independently in this distributed framework. Lastly, results of simulations with hundreds of truck in the Swedish transportation system shows that our method works well as expected, by which trucks are forming platoons effectively and are able to find out their optimal utilities.

Development of this distributed MPC framework can be further improved in different directions. One of the most useful way will be the adaption to the EU driver constrain, that the driver are nor allowed to be driving over 4.5 hours continually without taking a rest.

ACKNOWLEDGMENT

The authors would like to thank our supervisor Ting Bai for her help.

REFERENCES

- [1] S. Tsugawa, S. Jeschke, and S. E. Shladover, "A review of truck platooning projects for energy savings," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 68–77, 2016.
- [2] European Parliament. (2019, May) Co2 emissions from cars: facts and figures. [Online]. Available: <https://www.europarl.europa.eu/news/en/headlines/society/20190313STO31218/co2-emissions-from-cars-facts-and-figures-infographics>
- [3] Baptiste Chatain. (2019, Apr.) Meps approve new co2 emissions limits for trucks. [Online]. Available: <https://www.europarl.europa.eu/news/en/press-room/20190412IPR39009/meps-approve-new-co2-emissions-limits-for-trucks>
- [4] M. Saeednia and M. Menendez, "A consensus-based algorithm for truck platooning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 2, pp. 404–415, 2017.
- [5] E. van Nunen, F. Esposto, A. K. Saberi, and J.-P. Paardekooper, "Evaluation of safety indicators for truck platooning," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1013–1018.
- [6] W. Zhang, E. Jenelius, and X. Ma, "Freight transport platoon coordination and departure time scheduling under travel time uncertainty," *Transportation Research Part E: Logistics and Transportation Review*, vol. 98, pp. 1–23, 2017.
- [7] M. Saeednia and M. Menendez, "A consensus-based algorithm for truck platooning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 2, pp. 404–415, 2016.
- [8] M. Saeednia and M. Menendez, "Analysis of strategies for truck platooning: Hybrid strategy," *Transportation Research Record*, vol. 2547, no. 1, pp. 41–48, 2016.
- [9] V. Sokolov, J. Larson, T. Munson, J. Auld, and D. Karbowski, "Maximization of platoon formation through centralized routing and departure time coordination," *Transportation Research Record*, vol. 2667, no. 1, pp. 10–16, 2017.
- [10] R. Larsen, J. Rich, and T. K. Rasmussen, "Hub-based truck platooning: Potentials and profitability," *Transportation Research Part E: Logistics and Transportation Review*, vol. 127, pp. 249–264, 2019.
- [11] S. Van De Hoef, K. H. Johansson, and D. V. Dimarogonas, "Fuel-optimal centralized coordination of truck platooning based on shortest paths," in *2015 american control conference (acc)*. IEEE, 2015, pp. 3740–3745.
- [12] T. Bai, A. Johansson, K. H. Johansson, and J. Mårtensson, "Event-triggered distributed model predictive control for platoon coordination at hubs in a transport system," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 1198–1204.
- [13] C. Bonnet and H. Fritz, "Fuel consumption reduction in a platoon: Experimental results with two electronically coupled trucks at close spacing," SAE technical paper, DaimlerChrysler AG, Research Department, Stuttgart, Germany, Tech. Rep., 2000.
- [14] E. Stegner, J. Ward, J. Siefert, M. Hoffman, and D. M. Bevly, "Experimental fuel consumption results from a heterogeneous four-truck platoon," American Center for Mobility, USDOE Office of Energy Efficiency and Renewable Energy (EERE), Tech. Rep., 2021.

Adaptive Cruise Control and Platooning With Tire Slip Awareness

Gustaf Reimer and Filip Henriksson

Abstract—Platooning is a method where a chain of vehicles drive with small inter-vehicular distances. The many benefits of autonomous platooning includes improved fuel economy, less congestion and safer transportation. To create a safe and functional platoon the operational software needs to be able to handle various road surfaces without the risk of a crash. This report is aiming to improve the safety of a platoon by including communication of data between vehicles in the chain. Specifically the focus has been on transferring information about the tire slip, to model a cooperative adaptive cruise control (C-ACC) and combine the two. A system was designed using the dynamics for a quarter-car model and then connected to a controller and a platoon of four vehicles. Simulations of when the leading vehicle braked hard on two different road surfaces with and without the slip awareness was conducted. The tire slip awareness in the controller consisted of proportional control on the error and a low-pass filter. The simulations showed that the inclusion of the tire slip in the controller improved the platooning performance, in the sense that the inter-vehicle distance could be contained. It was also shown the controller could be tuned so that the slip ratios were limited.

Sammanfattning—Konvojkörning är en metod där en kedja av fordon åker med små interna distanser. De många fördelarna med förarlösa konvojer inkluderar förbättrad bränsle förbrukning, mindre trafik och säkrare transporter. För att en säker och funktionell konvoj ska kunna skapas krävs det att mjukvaran kan handskas med varierande vägunderlag utan risk att krocka. Den här rapporten siktar på att förbättra säkerheten i konvojkörning genom att överföra data till andra fordon i konvojkedjan. Speciellt har fokuset legat på överföra information om däckslirning, att modellera en kooperativ adaptiv farthållare (C-ACC) och sedan kombinera de två. Ett system designades genom att använda dynamiken av en fjärdedels bil och sen ansluta modellen till en konvoj med fyra fordon. Simulationer av när det ledande fordonet tvärbromsade på olika vägunderlag med och utan däckslirningsinformation genomfördes. Däckslirnings i regulatören bestod av proportionerlig kontroll på felet och ett lågpasfilter. Simulationerna visade att inkluderingen av däckslirningsinformation i regulatören förbättrar konvojens prestanda, på så sätt att de interna distanserna kan hanteras. Det kunde också påvisas att kontrollern kunde kalibreras så att slirningen begränsades.

Index Terms—Platooning, tire slip, car following, C-ACC, autonomous driving.

Supervisors: Jonas Mårtensson, Eshan Hashemi

TRITA number: TRITA-EECS-EX-2022:123

I. INTRODUCTION

The automation of vehicles will lead the automotive industry into a new era with safer roads and more efficient transportation methods. According to [1] 90 % of crashes in traffic happen due to human errors. Creating autonomous

platoons is a natural early transition to achieve the benefits of automation by eliminating the human factor. Platooning is when vehicles, often trucks, travel with small inter-vehicular distances to improve fuel economy and decrease allocated space. In this project the group will develop a platoon model with tire-slip awareness to improve safety within it.

A. Background

The main benefit of platooning is the reduced air resistance for the following vehicles. Platooning is most efficient for heavy duty vehicles such as trucks on highways, but the principle works for all different kinds of transportation systems. For chains of heavy duty vehicles air resistance can be reduced by up to 40% at highway speeds for the followers [2], and lower fuel consumption by more than 10% [3]. To reach these numbers the longitudinal distances need to be short. Driving vehicles at 80 *kph* and above with spacing of only five to ten meters seems like a operation deemed for failure and potentially extreme consequences. To make platooning a reality these vehicles have to be operated not by a human, but by software to control velocities, acceleration and spacing. Adaptive cruise controllers (ACC) is nowadays fairly common in commercial cars where the system reacts to information obtained from sensors and cameras which scan the surrounding area. ACCs could be used for platooning purposes but can not ensure string stability according to [4]. String stability can be described in various ways, for example as in [5], where it's defined as the amplification of error in relative distance, speed or acceleration further down in the platoon. If the amplification increase uncontrollably, the system is considered string unstable.

However a future possibility for a string stable platoon would be if the vehicles used a cooperative adaptive cruise control (C-ACC) where information such as speed, position and other relevant information is communicated wirelessly to other members in the string. This will make the system react faster and more accurately to the other vehicles.

B. Problem formulation

Platooning with short inter-vehicle distances is feasible in favorable road conditions but what if the road is full of snow or ice? When the traction is low, it seems rather dangerous to travel with as little as 8 meters in between if the front vehicle suddenly has to panic brake. To tackle this problem the project group will investigate if using the communicated tire slip ratio from the preceding vehicle could improve the performance of the whole platoon. Will including the tire slip

in the cruise controller improve the platooning performance? Can information about the tire slip create safer and smarter platoons?

C. Related works

The project group has not come across any related works where the tire slip is used in the controller of either a single vehicle cruise controller or in a platoon. This project is interesting because it explores if the performance and safety of a platoon is improvable by including the tire slip in communication. There are however plenty of works that deal with cruise control for a platoon. As shown in [6] the ACC:s available on the market as recent as 2018 are string unstable when put in a platoon. This indicates that further connectivity beyond what the vehicles can detect themselves with radar is necessary for platooning to work.

D. Structure

This report is divided into five main sections. First the introduction and background. Section II describes the underlying physics and equations used as well as the problem definition. In section III the controller design is described, section IV contains the result, V the discussion and finally conclusion in section VI.

II. PRELIMINARIES AND PROBLEM DEFINITION

A. Dynamics

To understand the approach of the thesis one needs to be familiar with some basic vehicle dynamics. Shown in figure 1 is the longitudinal dynamics of a car.

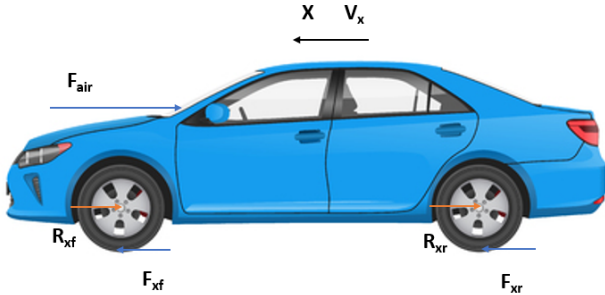


Fig. 1. Longitudinal vehicle dynamics of a car

The forces acting on the car is F_{air} an aerodynamic drag force, R_{xf} and R_{xr} which is rolling resistance on the front and rear tires, and longitudinal tire forces F_{xf} and F_{xr} . From these forces the following equation can be put together

$$m\ddot{x} = F_{xf} + F_{xr} - F_{air} - R_{xf} - R_{xr} \quad (1)$$

where m is the mass of the car and \ddot{x} is acceleration according to Newtons second law of motion. The rolling resistance is

$$R_{xi} = fmg \quad (2)$$

where f is a rolling resistance coefficient and g is the gravity. The air resistance is

$$F_{air} = \frac{1}{2}cA\rho v_x^2 \quad (3)$$

where c is drag coefficient and A is the frontal area of the vehicle. The group opted to use a quarter car model instead

of the entire car model to simplify the problem. In figure 2 the dynamics of a lone wheel is shown

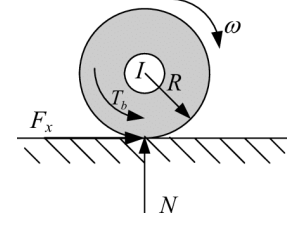


Fig. 2. Dynamics of a wheel [7]

where R is the tire radius, N is the normal force and ω is the angular velocity. F_x is the same tire force as seen in figure 1, but for one specific wheel. An equation can be obtained

$$\dot{\omega} = \frac{T - RF_x}{I} \quad (4)$$

where $\dot{\omega}$ is angular acceleration, I is moment of inertia of the tire and T is the applied torque from the engine on one wheel. The angular velocity is measurable by sensors and considered known in this case. The slip of the tire is a ratio and described as

$$\lambda = \frac{R\omega - V_x}{\max(R\omega, V_x)} \quad (5)$$

where V_x is the longitudinal velocity of the vehicle. Everything in (5) is established. If V_x is greater than $R\omega$ the vehicle is braking and vice versa. If the wheels lock up then $\omega = 0$ which means that $\lambda = -1$. The tire force is wholly dependent on the slip. This means that a certain section of the contact patch between the ground and the tire has to experience slip for there to be a tire force at all. Dependence on the deformity of the contact patch and other factors means that this relationship is pretty complicated.

$$F_x = mg \cdot D \cdot \sin(C \cdot \arctan(B \cdot \lambda - E[B \cdot \lambda - \arctan(B \cdot \lambda)])) \quad (6)$$

In (6) the correlation between the tire-slip ratio, λ , and the road friction coefficient, μ , is described according to the so called magic formula, where B , C , D and E are surface dependant constant. The magic formula is derived from $F_x = F_z \mu$ where F_z is the normal force on the wheel from the surface. When the slip is small the function is approximately linear, but around 0.1 it begins to flatten out, reaching a peak value around 0.15 where the traction is the best, before seeing a slight drop, mainly for the higher μ values, as is described in [4]. In figure 3 the peak friction coefficient, μ , for different surfaces is displayed according to the magic formula [8]. The peak values for each line is when the slip ratio is around 0.1 – 0.2, which is the preferred value of tire slip, meaning that the maximum force is obtained in that region. The y-axis in figure 3 can be interpreted as normalized force.

The final system of dynamics, with (1)-(6) combined, is displayed in figure 4 where torque is used as input controlled by a cruise control.

B. Platoon

In figure 5 the information sharing in a platoon is shown. The preceding vehicle's position, velocity and acceleration is normally the kind of communicated information. As seen in

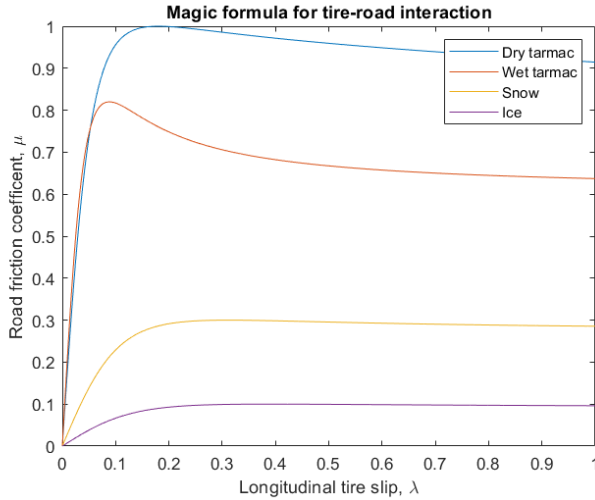


Fig. 3. A graph of friction coefficient as function of longitudinal slip for different surfaces using magic formula, [8].

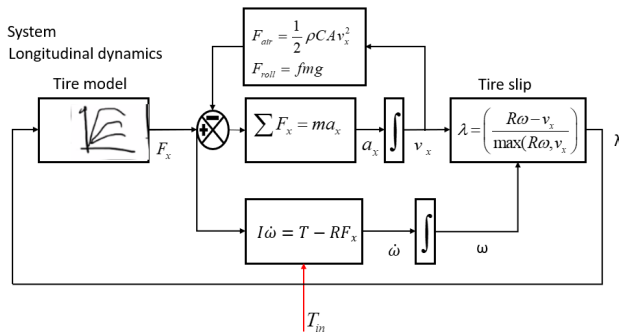


Fig. 4. The longitudinal vehicle dynamics with torque, T_{in} as input.

figure 5 the vehicles need to drive closely to attain the benefits of platooning while at the same time not risk a collision.

In this project the we have included the tire slip into the shared information to hopefully improve the platoon in bad road conditions. Since we have not seen any project like this before, there is no direct answer.

C. Control

To form a viable and functional platoon with rather slim inter-vehicle distances the controller needs to be up to a certain standard. Some important criteriums is to be able to maintain set distance and react to changes in speed without jeopardising comfort and safety. Although the optimization

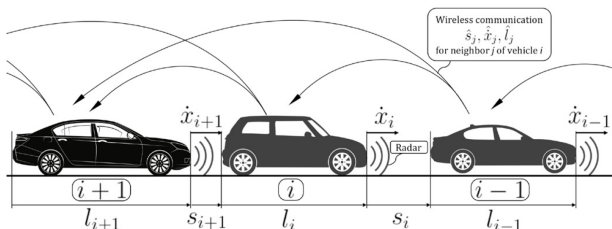


Fig. 5. Information sharing in a platoon [9].

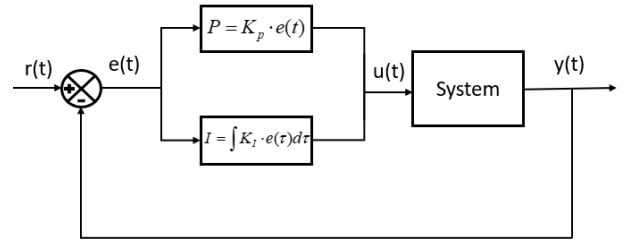


Fig. 6. A PI-controller.

of the controller was not the main task in this project, a lot of time was spent developing it. The project group opted for a standard proportional–integral–controller (PI) as seen in figure 6. This controller uses the gains to drive the given errors to zero. Depending on the relationship of the gains, the controller can be tuned to prioritize certain errors. This will effect the overall response of the system. The problem with using only proportional control is that this can result in static error, meaning for example that if the desired spacing is 10 meters, the proportional controller may only be able to get the spacing to 11 meters. To solve this integral action, I part in figure 6, is needed. However, since the acceleration and the velocity is the derivative and double derivative of the position, integral velocity control is the same as proportional position control and so forth. This means that PI-control is only necessary on the position error, while P-control suffices for the the velocity and acceleration errors.

D. Problem definition

When only using the distance, velocity and acceleration measurements in the controller there is a risk of misunderstandings within the platoon. Low tire capacity might be misinterpreted for a mild deceleration, causing the platoon to become unstable. Besides causing unwanted disturbances within the platoon, the slip in this case goes beyond the saturation point, meaning that the vehicle will become difficult to steer. To keep this from happening the project group attempted to use the slip in the controller to more fully communicate the situation, reduce disturbances and keep the slip from going past 10-20% for trailing platoon members.

In terms of limiting the slip when braking, it is true that most vehicles already have an Anti-lock braking system (ABS) tasked with doing just that, but this system only makes sure that the wheels don't lock up, not necessarily that the slip remains around the saturation point. The goal is to limit the tire-slip as much as possible for the trailing vehicles to maintain handling and improve the safety of the platoon.

III. CONTROLLER DESIGN

While the leading vehicle followed a set trajectory the trailing vehicles attempted to maintain the same velocity and acceleration at a safe inter vehicular distance with a combination P- and PI-controller, see (7).

$$T_{i,noslip} = (C_1 + \frac{C_2}{s})(x_{i-1} - x_i - l) + C_3(v_{i-1} - v_i) + C_4(a_{i-1} - a_i) \quad (7)$$

Where x is position, v velocity, a acceleration and l spacing. The index i is the ego vehicle and $i - 1$ is referring to the preceding one, as seen in figure 5. This controller, (7), was then augmented by a P-controller using the slip

$$T_{i,slip} = T_{i,noslip} + C_5(\lambda_{i-1} - \lambda_i) \quad (8)$$

in order to evaluate if adding the tire slip term can improve the performance on slippery road conditions. The idea behind 8 was that it could potentially subdue the unbounded and unstable behaviour a controller expecting high tire capacity would experience on a surface such as snow. Finally a third controller were proposed with the same shape as (8) but with an added low-pass filter.

$$T_{i,bothslip} = T_{i,noslip} + C_5(\lambda_{i-1} - \lambda_i) - C_6 \frac{\lambda_{i-1}}{\tau s + 1} \quad (9)$$

The idea behind (9) was that the low pass filter would work oppositely the requested torque, meaning that when the torque and slip got too negative the low pass filter would add some positive torque in order to ease the braking and decrease the slip. This could theoretically have been done by simply feed forwarding $-C_6\lambda_{i-1}$, but by having it in a low pass filter instead makes the controller more stable, since the time constant τ , delays the process and makes for less volatility.

Tuning a controller with many gains such as this one seen in (8) and (9) is time consuming and difficult. Since there is cohesion between the gains and a change to one will alter the effect of the others. In the end a satisfactory controller were tuned, and the results is seen in the next section, section IV.

IV. RESULTS

A. Simulation setup

The simulation was setup for the leading vehicle to brake from 30 m/s to 10 m/s while 3 trailing vehicles tried to maintain an inter-vehicular spacing of 10 meters. The system was implemented into MATLAB's Simulink for simulation purposes, an overview of the platoon can be viewed in appendix A. The leading vehicle used a cruise control (PI-control) for velocity changes, the values is found in appendix B. The three trailing vehicles were all equipped with the same C-ACCs, where velocity, position, acceleration and eventually tire slip from the preceding vehicle were communicated. In table I the common data for all platoon members is displayed.

TABLE I
PARAMETER DATA FOR SIMULATIONS.

Parameter	Value	Unit
Tire radius, R	0.31	m
Mass, m	450	kg
Gravity, g	9.81	m/s^2
Normal force, N	4414.5	N
Moment of inertia, I	1.5	kgm^2
Rolling resistance coefficient, f	0.08	-
Air density, ρ	1.225	kg/m^3
Frontal area, A	2.04	m^2
Drag coefficient, C	0.4	-

The simulations had four different setups:

- Dry tarmac without slip term.

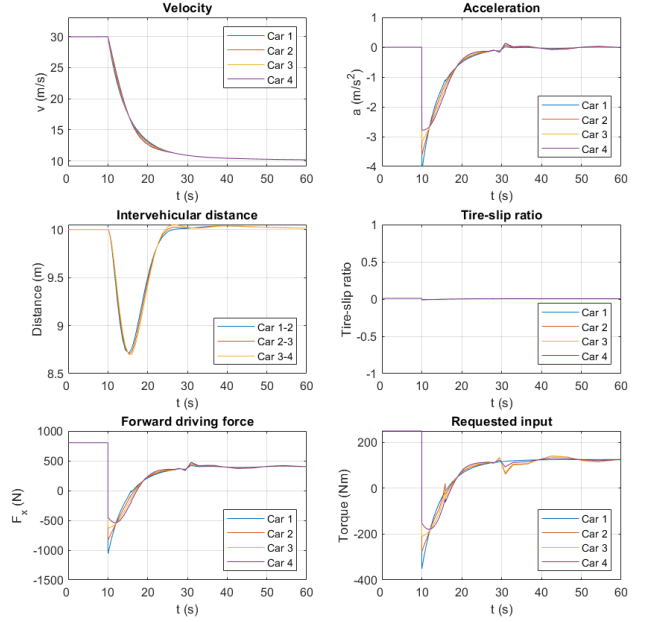


Fig. 7. Platoon driving on dry tarmac (max $\mu = 1$) without tire slip terms in their controllers.

- Changing road surface at $t = 10s$, from dry tarmac to snow without slip term.
- Changing road surface at $t = 10s$, from dry tarmac to snow with additive slip term.
- Changing road surface at $t = 10s$, from dry tarmac to snow with both additive slip term and low-pass filter.

The tire constants in (6) used for the two different surfaces is displayed in table II, according to [8].

TABLE II
MAGIC FORMULA CONSTANTS FOR DRY TARMAC AND SNOW

-	B	C	D	E
Dry tarmac	10	1.9	1	0.97
Snow	5	2	0.3	1

These constants in implemented into (6) form the blue and the yellow curve in figure 3. All controllers were simulated in these four setups to make sure that there was a discrepancy between low traction and high traction without slip and that the inclusion of the slip could diminish this discrepancy.

B. Controller without slip

That the platoon would experience difficulties when trying to brake on a slippery surface seemed pretty obvious, but what would go wrong and how this would look exactly was something that the model would have to show. As can be seen in the figure 7, with high traction the platoon in this tuning works fine. The controller comes from (7) where the constants C_1 to C_4 values is found in appendix B. The intervehicular distances does not go down too far to risk the stability. The input graph down on the right shows that the

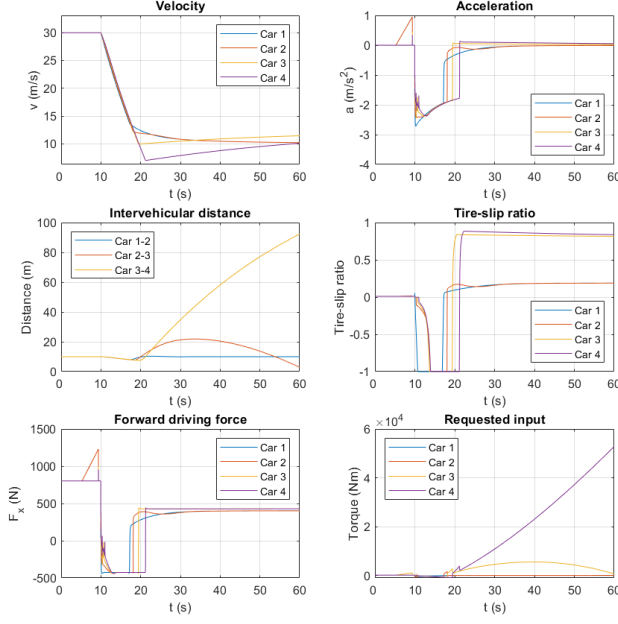


Fig. 8. Platoon driving on snow ($\max \mu = 0.3$) without tire slip terms in their controllers.

requested torque is around $-300Nm$, which is reasonable, and the vehicles easily attain $-1000N$ without encountering any limitations, making it easy for them to brake properly. While in the case of low tire capacity using the same controller as in figure 7, the control input grows more and more unbounded for every vehicle seen in figure 8. Because the controller is tuned to expect dry road traction, the vehicles' response to the maximum possible force being severely limited, at about $-500N$, is to ask for even more force. This means that not only do the slips reach -1 and remain so for a longer and longer period of time, they actually shoot up to around 0.75 after the initial brake, causing some very volatile behavior. As seen on the inter vehicular distance graph in figure 8 the last vehicle then falls away and loses connection to the platoon. The third vehicle collides into the back of the second vehicle as well. If a real platoon acted this way the C-ACC would be forced to disengage. Notice that the acceleration reaches only about $-2.7 m/s^2$, what would be considered a mild deceleration when on dry tarmac. Receiving this number from the lead vehicle the second vehicle has no way of knowing of the danger ahead, until the inter vehicular distances grow out of control and the controller overcompensates.

C. Controller with slip error P-control

Introducing the slip term, the system became a lot more stable, see figure 9. Tuned properly using the values in appendix B for (8), it was possible to make the vehicles experience less and less slip while keeping inter vehicular distances comparable to the controller without slip on dry tarmac. Analysing the graphs, the key seemed to be a faster reaction time. Because the rest of the slips adhere more closely to the shape of the lead vehicles slip the trailing vehicles begin

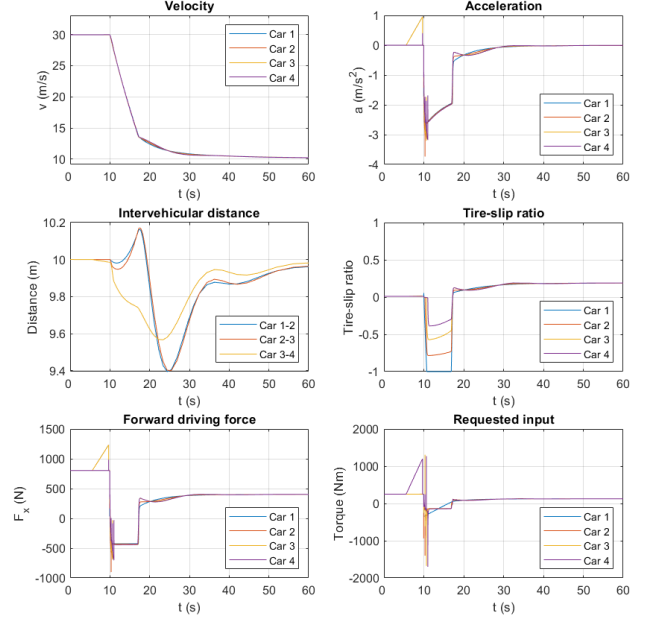


Fig. 9. Platoon driving on snow ($\max \mu = 0.3$) with additive tire slip term in their controllers.

to break when they're still on the dry tarmac. This is revealed by the spikes at around 10 seconds in the acceleration and force graphs. Braking on the dry tarmac the vehicles manage to attain $-4 m/s^2$ acceleration briefly, before encountering the snow, making for a much smoother brake. That the tire slip decreases for each vehicle makes a difference for handling and available brake force as seen in figure 3.

D. Controller with both slip error P-control and low-pass filter

For the last simulation the controller from (9) is used with tuning values found in appendix B. As seen in 10, this controller improve on the results from the previous simulations as far as reducing the tire slip for the trailing vehicle goes. This controller handles the spacing between the vehicles worse than the controller with just the additive slip term. As seen in figure 10 the distance between the third and fourth vehicle is below nine meters for a short period of time. Although this is nothing serious in this context, it is worth noting.

V. DISCUSSION

A. If friction was measurable

The friction coefficient is not easy to measure or estimate which complicates this problem. If the leading vehicle would be able to measure the friction of the road surface, it could communicate it to the platoon and repercussions for inter-vehicle distances and speed would take place. There is a reason why big road transportation companies like to get their hands on such technology. With such technology the whole platoon would know the second the road surface changes and either adapt their velocity or increase their spacing to a safer distance.

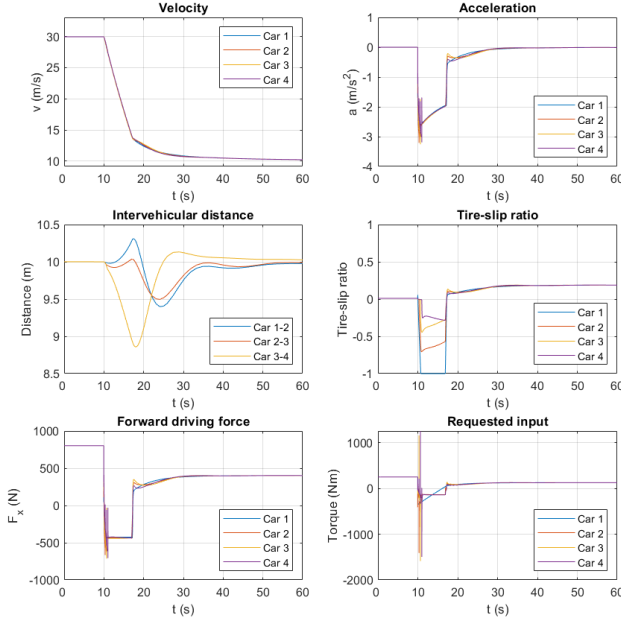


Fig. 10. Platoon driving on snow ($\max \mu = 0.3$) with both additive tire slip term and low-pass filter in their controllers.

B. Tuning

With the $T_{i,noslip}$ controller tuned more optimally for the dry road surface, the adding of the slip term was not as effective at limiting the slip. In almost every tuning that was tried the inclusion of the slip term always had a positive effect on the stability and inter vehicular distances, but for the slips to also be limited, the tuning had to be more specific. The added slip term in itself does nothing to limit the slip, it actually tries to do the opposite if the lead vehicle slip reaches -1 . It relies on the overall tuning to limit the slip. With the basic tools of error reduction and feed-forward low pass filter, it was not possible to create a control law tasked with keeping the slip around 0.1-0.15 slip, but it was possible to tune it to reduce the slip for trailing members.

C. Slip as the added term

Adding more connectivity and information to the platoon's communication will almost invariably increase the performance, no matter if it's slip, angular velocity or velocity from more vehicles. This should be kept in mind when evaluation the improvements of including the slip. Also the difficulty of measuring slip and disturbances one may encounter has not been taken into account in the simulations.

D. Slip in the low-pass filter

With the $T_{i,noslip}$ controller the vehicles do not realize that at a certain point requesting more torque will not get them more tire force to work with. Finding a way to make them realize this was difficult. The slip-limiting effect of adding the low pass filter was not very big, but as opposed to slip P-controller it had the right idea behind it, to try and hold back the torque if the slip became large.

E. Simulation bugs

In figures 8 and 9 the driving force, acceleration and requested torque spike noticeably before $t = 10$ s. As the first vehicle has neither encountered the snow or begun to break at this point and as the time gap between the two points are around four seconds this is regarded as a bug in the solver.

F. Further work

There are aspects in this project that have the potential to be developed further. The C-ACC has a potential to be improvement for autonomous platoons in the future. This is a relative basic one and can be developed into a better one, for example a proportional-integral-derivative-controller (PID). One could spend a long time tuning the controllers further to optimize the performance. Another option is to use a model predictive controller (MPC) together with optimization theory to increase the performance further.

The model could also be more dynamic and developed to handle three dimensions. For example would more realistic tire formulas help to improve the accuracy of the simulations. Lateral and vertical vehicle dynamics could help broaden the project and lead to a more complete package.

Inclusion of disturbances such as latency in communication and hard blowing winds could be interesting to evaluate. How would the platoon react to these kinds of things while maintaining spacing accordingly.

VI. CONCLUSION

To conclude, the inclusion of the tire slip in the controller of the vehicles in the platoon is improving the performance for a slippery road surfaces. However the controller is not ideal and could certainly be improved upon to handle more variations in condition. Different type of controller can be evaluated to for different purposes but the current one is a good first step.

ACKNOWLEDGMENT

The authors would like to thank the supervisors Jonas Mårtensson and Eshan Hashemi for their support and dedication to this project.

REFERENCES

- [1] (2008, Jul) National motor vehicle crash causation survey - transportation. [Online]. Available: <https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/811059>
- [2] A. Alam, B. Besselink, V. Turri, J. Mårtensson, and K. H. Johansson, "Heavy-duty vehicle platooning for sustainable freight transportation: A cooperative method to enhance safety and efficiency," *IEEE Control Systems Magazine*, vol. 35, no. 6, pp. 34–56, Dec. 2015.
- [3] *Fuel Consumption Reduction in a Platoon: Experimental Results with two Electronically Coupled Trucks at Close Spacing*, ser. SAE Technical Paper Series. Costa Mesa, California: Society of Automotive Engineers, Inc, 2000.
- [4] R. Rajamani, *Vehicle Dynamics and Control*, 2nd ed., ser. Mechanical Engineering Series. Minneapolis: Springer, 2012.

- [5] J. Ploeg, N. van de Wouw, and H. Nijmeijer, “Lp string stability of cascaded systems: Application to vehicle platooning,” *IEEE transactions on control systems technology*, vol. 22, no. 2, pp. 786–793, Mar. 2014.
- [6] G. Gunter, D. Gloudemans, R. E. Stern, S. McQuade, R. Bhadani, M. Bunting, M. L. D. Monache, R. Lysecky, B. Seibold, J. Sprinkle, B. Piccoli, and D. B. Work, “Are commercially implemented adaptive cruise control systems string stable?” *arxiv.org*, Jul. 2019.
- [7] Y. Jing, Y.-e. Mao, G. Dimirovski, Y. Zheng, and S. Zhang, “Adaptive global sliding mode control strategy for the vehicle antilock braking systems.” *IEEE*, Jul. 2009, pp. 769 – 773.
- [8] (2022, Apr.) Tire-road interaction (magic formula). [Online]. Available: https://www.mathworks.com/help/physmod/sdl/ref/tireroadinteractionmagicformula.html#bso7k1e_vh
- [9] 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, Mar. 2019.

CONTEXT B

AUTONOMOUS ROBOTIC SYSTEMS

POPULAR DESCRIPTION

How to Not Create a Terminator

Have you ever thought about how robots make decisions? Or what makes a robot trustworthy? This field is rapidly progressing into our everyday life and needs to be handled cautiously. If an engineer fails to program the robot correctly, the consequence could be “death by robots”.

When people think about the dangers of an automated world, many think of movies like The Terminator, where humans are hunted by evil robots that are designed to exterminate us. Evil robots aren't a threat to people's safety in reality, but stupid robots might be. As the late pastor Dietrich Bonhoeffer once said: “Stupidity is a more dangerous enemy of the good than malice.”. An evil robot will soon be caught and destroyed, but a poorly programmed robot might go unnoticed while working as intended – until suddenly, disaster strikes!

To avoid the mentioned mishappens, the robot needs to be programmed correctly. For instance, how does an autonomous robot know not to collide with other robots, or more importantly, with humans? It's the engineers that put in lines of code into the robot, which makes it avoid obstacles and helps it handle collisions. For example, if you fall asleep on your lawn while your robotic lawn mower is on. What will happen? It will turn away from you, since it's been programmed to avoid obstacles, assuming the engineers weren't sloppy – otherwise the consequences could be fatal. That is why it's important to continue investing in robotic systems.

To save mankind, we, the engineers, will prevent robots from turning on humans. We will control them and program them to protect us. We will make them our friends. Therefore, it's reasonable to say that: We are the heroes of our time, but we're dancing with the robots in our minds.

SUMMARY OF PROJECT RESULTS

Autonomous artefacts are becoming increasingly more popular in today's society. It is no longer important from only an industrial perspective, but also in everyday life. Its area of application varies from self-driving cars, unmanned aerial vehicles to robot vacuum cleaners, among other things.

The main advantage of autonomous devices is their capability to replace or improve human's ability to handle and manage complex tasks, which signifies analysing and making decisions based on data under real-time. Therefore, these systems also need to cooperate with both humans and with themselves in order to solve complex tasks. The consequence of this, is the rise of new trends regarding collaboration and interaction of autonomous systems, which is what this context revolves around.

Project group B1 had the goal of implementing a motion planning algorithm, which would result in a collision free path between one initial and one goal state. Project group B2, on the other hand, had the purpose of moving a group of autonomous ground vehicles through a predetermined path, while dynamically avoiding obstacles. Lastly, project group B3 aimed to make a group of agents, meaning robots, complete timed tasks sequentially, while avoiding collisions.

In project B1, the group aimed to develop an understanding for the algorithms behind the trajectory of an Unmanned Aerial Vehicle, UAV. This was done by analysing one method of motion planning algorithm and testing the method by using various

simulation environments, which consisted of fixed obstacles. The experiment is considered successful if the UAV - in terms of a quadrotor - can, based on one initial and one goal position, generate a feasible path – meaning taking the quadrotor dynamics into consideration while also avoiding obstacles. The main idea of the project was to develop a way of changing the intermediate points between the initial and final position, in order to create the shortest path possible – meaning minimizing a cost function - while not intersecting with obstacles.

One large improvement for project B1 would be for the group to use optimal trajectory generation. This would make a significant reduction in the computational cost, and exclude the need of aggressive manoeuvres, which in turn would allow for the creation of a more computationally efficient obstacle avoidance algorithm. One further improvement would be for the algorithms to also handle moving objects with unknown locations since it would make the UAV functional in real life.

In project B2 the group studied how to move a formation of autonomous ground vehicles (AGV:s) along a determined path while avoiding obstacles placed along the path. This was achieved by solving and combining three individual control problems: reference tracking, formation control and obstacles avoidance. The stability of the model was proven by simulating the model with MATLAB. In the simulation a formation of 5 AGV: s was driven along a 2-dimensional path filled with static obstacles modelled as points with potential fields.

Possible future improvements on the project could be adapting the model to a more complicated and realistic driving system, simulating more generic obstacles such as walls and testing the simulated models with physical AGV: s. Further extensions of the project could be to expand the model into 3-dimensional space with aerial vehicles or topological mapping.

Project group B3 designed a strategy for making groups of agents execute timed tasks in a specified sequence while avoiding collisions with other agents and obstacles. The tasks are described using a subset of Signal Temporal Logic, STL. STL is a logical language used to describe time dependent actions, such as staying within a specific area during a given time frame, in a machine friendly manner. Control Barrier Functions, CBFs, which is a tool for designing control-based strategies, is used to ensure that the tasks are performed as specified. The group used a baking recipe as the sequence of tasks to be executed in a bakery environment.

One future use for the project is to compare CBF based control for timed sequential tasks to alternatives such as Mixed Integer Linear Programming, MILP, since CBFs introduce a number of constraints on the STL specifications.

IMPACT ON SOCIETY AND ENVIRONMENT

From an individual perspective, autonomous systems offer many advantages that simplify everyday living - like robot vacuum cleaners. The area also offers a lot of potential for the future. For example, autonomous vehicles could provide a safer way of transportation, and prevent accidents that are caused by humans. We should also mention the fact that autonomous systems are strongly connected with our economy. There is even a trend, within companies, to introduce autonomous system in order to maximize efficiency and increase profits. This consequently leads to a decrease of simple jobs. Superficially, this would mean less jobs, but that is not necessarily true. It would instead be a movement of available jobs. Robots should be used as tools to help us make our jobs and lives easier, not replace us. Instead of cleaning by ourselves we would supervise robots cleaning or evaluate how well the robots are doing their jobs. A more autonomous society would lead to a stronger economy and jobs not taken by robots but integrated with robots.

There are however some important disadvantages regarding how people without engineering degrees would fit in. They would not be able to contribute to a society where many human tasks are being done by robots since they do not have enough knowledge about robots to make them useful. There would not be any jobs available for them, meaning they would not be able to make a living nor feel connected to the society. The consequence of this could be, on an individual level, feeling a loss of purpose. Therefore, it is important to maintain robots as tools only and to make sure that people in the future also have a way of contributing to society. To achieve this, one might need to redesign all education, from elementary school to high school, and to create new jobs that require the competence of the people that otherwise would have been excluded. The result would be that that society develops in a way that makes life safer for the individual. Robots could perform dangerous but necessary tasks, while humans focus on new tasks.

Another consequence of a technically advanced society is the fact that everything is stored digitally, which means that a lot of personal information is being collected. We do not know how this information is being collected and used, which makes us susceptible to all kinds of advertising that uses this. This could, depending on the individual, be very damaging. Not to mention the fact that the information, in the wrong hands, could be used to repress people and their human rights, which is something that we can see happening in other countries. Therefore, even though a more technical world would be safer for the individual, it would not be ethically justified, due to the lack of personal integrity. With this in mind, we believe it to be an ethical dilemma regarding safety vs personal integrity, where the latter is considered to be more important.

There are many aspects to look at when it comes to autonomous robotic systems and society. If a robot stops working while in the middle of operating on a human and because of that the person dies, who will get the blame in a legal process? Is it the robot, the hospital, the programmer, or the engineer that made the robot in the first place or maybe even the person who installed it in the operating room? Perhaps this and every similar case with robots will be written off as an accident. Will that then lead to a future where robots can make all kinds of mistakes without anyone getting the blame for it? We creators need to put some thought into how to handle our creations.

Another ethical aspect is about robots used in military operations. Is deploying autonomous robots in warfare morally justifiable or not? On one hand autonomous robots might make perfect killing machines absent of both morality and pain, capable of making and executing efficient and cold-hearted choices in order to win. On the other hand, these perfect killing machines might just be deployed to fight other such machines, which would significantly reduce human suffering on the battlefield. Fighting using robots would transition wars into an economic battle instead of a bloody one, not ideal but a better alternative than needlessly throwing humans lives at each other.

Autonomous systems can also help make our cities and communities more sustainable by optimizing public transportation systems, power grid systems, traffic etc. Examples of this could be autonomous subway trains that accelerate and decelerate as energy efficiently as possible or autonomous cars driving environmentally friendly. Automation is used for aiding the process of recycling and could be further developed to make it easier for people to recycle waste. Recycling stations are not easily accessible for everybody, and people also need to be incentivised to recycle since it is very convenient to just throw all your trash in one place. If there existed a recycling system that automatically recycled all the waste that gets thrown into it, these problems would be solved, even if it might not be the most economically sound method.

Even if all the changes mentioned above were implemented successfully, the result would only be a reduction in the negative impact of current systems on the environment. In the opinion of the authors, the greatest strengths of automation are economic benefits and benefits of convenience rather than environmental benefits. For that reason, perhaps the most important part of designing autonomous systems with regard for the environment is not the ways in which we can use them to directly improve our environment, but rather how we design our systems to minimize their negative impact on the environment.

Motion Planning for Aggressive Flights of an Unmanned Aerial Vehicle

Filippa Femic and Cornelia Smith

Abstract—Unmanned aerial vehicles are becoming more popular in today's society, which results in the rise of laws intended to maintain safety. To abide by these, while allowing the technology to expand, functioning path-planning algorithms are required. This also includes having methods for detecting and managing obstacles. This project aims to improve an existing path-planning algorithm that is based on A* and implemented in Python. The solution consisted of using functions for finding polytope-intersection, as well as optimizing the collision avoidance and the search-algorithm. In addition to that, realistic constraints were implemented on the generated trajectory in order to reflect real-life limitations. The results demonstrated that the paths were always feasible, with respect to input and position constraints. The program's computation time was also reduced up to 89% of the original run-time. There is, however, still room for improvement since the original code generated a shorter path for the three scenarios it was created for. On the other hand, the improved algorithm could handle a new scenario, which the original code failed to do.

Sammanfattning—Obemannade flygfarkoster blir alltmer vanliga i dagens samhälle, vilket resulterar i uppkomsten av nya lagar ämnade åt att upprätthålla säkerhet. För att förhålla sig till dessa, samtidigt som teknologin tillåts expandera, krävs fungerande vägplaneringsalgoritmer. Där ingår det även att ha metoder för att upptäcka och hantera hinder. Detta projekt syftar till att förbättra en befintlig vägplaneringsalgoritm som är baserad på A* och implementerad i Python. Lösningemetoden bestod av att använda inbyggda Python-funktioner ämnade åt att finna skärningar mellan polytober, samt optimera kollisionshandling och sökalgoritmen. Dessutom infördes realistiska krav på den framställda vägen i syfte om att reflektera verklighetens begränsningar. Resultatet visade att vägarna alltid var genomförbara, med avseende på inmatnings- och positionsrelaterade villkor. Programmet beräkningstid hade även reducerats upptill 89% av den ursprungliga körtiden. Det finns dock utrymme för förbättringar då den ursprungliga koden generar en kortare väg för de tre scenarion den tillverkades för. Däremot kunde den förbättrade algoritmen hantera ett nytt scenario, vilket den ursprungliga koden misslyckades med.

Index Terms—UAV, autonomous vehicle, motion planning, polytope intersection, trajectory generation, obstacle handling, collision avoidance, feasibility constraints, differential flatness

Supervisor: Xiao Tan

TRITA number: TRITA-EECS-EX-2022:124

I. INTRODUCTION

Unmanned Aerial Vehicles (UAV), more commonly known as drones, consist of multiple individual rotors attached to a rigid frame. This report will focus on a quadrotor - meaning a UAV consisting of four rotors.

UAVs are becoming increasingly more popular in today's society. They have previously only been used for military

purposes, but lately expanded to commercial usage [1]. A natural consequence is the rise of laws and regulations in the EU, intended to maintain safety, for example not endangering others while flying nor taking images in certain areas [2]. This combined with autonomy - the ability to operate by itself [3] - results in more constraints on the drone. They are however still in an experimental stage and there is much room for improvement [1].

The purpose of this project is to get an understanding of a quadrotor. This includes learning how it operates and to create trajectories that depend on its dynamics, as well as the environment it travels in. One important aspect is obstacle handling, since it is what will prevent the UAV from colliding with objects, which can be tied to the regulation regarding endangering humans.

Another important aspects is the generation of a trajectory. This is done through the usage of path planning algorithms. They have the purpose of creating a path between a specified initial and final position, while often avoiding obstacles located in between. The optimal path, if one exists, will be the shortest distance between the positions, as well as having the lowest computational time. Additionally, the trajectory should not violate the UAV's motion constraints [4]. In an environment with obstacles, the algorithm should avoid collisions, while also taking the previous requirements into consideration. Example of such algorithms are A*, D* and Fast Marching, which are all graph-search algorithms [5].

This paper will focus on improving a trajectory generation algorithm that is based on A*, where the main focus will be on the collision handling, constraints to real-life limitations and the pathfinding algorithm.

II. UAV DYNAMICS

The UAV will perform movements from 3-dimensional states, defined by position, velocity, and acceleration. These will be affected by the thrust and motion generated by the rotors.

A. Dynamical definition

The dynamical model for a UAV, in terms of a quadrotor, is defined in [6]. Using this, two body-frames can be considered. The first frame, $\mathcal{F}_w = \{\vec{x}_w, \vec{y}_w, \vec{z}_w\}$, represents the inertial frame, while the second frame, $\mathcal{F}_b = \{\vec{x}_b, \vec{y}_b, \vec{z}_b\}$, corresponds to a frame fixed on the body with origin \mathcal{O}_b , see Figure 1. Following this setting, the parameters are defined as:

$m \in \mathbb{R}$ the UAV mass
 $v \in \mathbb{R}^3$ the velocity vector

$x \in \mathbb{R}^3$ the position vector with center of mass in \mathcal{F}_b
 $\Omega \in \mathbb{R}^3$ the angular velocity in the body-frame \mathcal{F}_b
 ${}^wR_b \in SO(3)$ the rotation matrix from \mathcal{F}_w to \mathcal{F}_b
 $f_i \in \mathbb{R}$ the thrust generated by the i :th rotor along $-\vec{z}_b$.
 $J \in \mathbb{R}^{3 \times 3}$ the inertial matrix of frame \mathcal{F}_w
 $f \in \mathbb{R}$ the sum of thrust magnitude
 $j \in \mathbb{R}$ the jerk
 $M \in \mathbb{R}^3$ the total momentum vector in \mathcal{F}_b

The sum of thrust magnitude is defined as $f = \sum f_i$ for $i \in \{1, 2, 3, 4\}$.

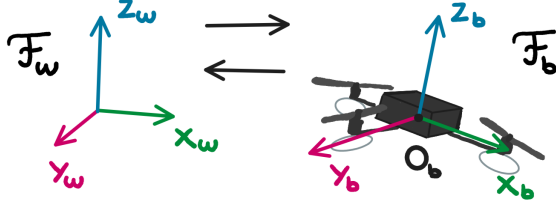


Fig. 1. The quadrotor's body-frame \mathcal{F}_b related to the inertial frame \mathcal{F}_w [7].

B. Dynamic Model

The motion of the quadrotor could, according to [6], be described by the following equations:

$$\dot{x} = v, \quad (1)$$

$$m\ddot{x} = m\dot{v} = mgz_w - {}^wR_b f z_w, \quad (2)$$

$${}^w\dot{R}_b = {}^wR_b [\Omega]_{\times}, \quad (3)$$

$$\dot{\Omega} + \Omega \times J\Omega = M, \quad (4)$$

where $[\Omega]_{\times}$ is the skew-symmetric matrix such that: $[\Omega]_{\times} = \Omega \times v$, where \times is the vector cross-product [8].

C. Differential Flatness

Differentially flat systems are systems for which we can find a set of outputs, with the same dimension of inputs, such that all states and inputs can be obtained from the outputs without integration. This is particularly useful in trajectory generation for quadrotors, since it results in explicit equations [9].

Using the equations in Section II-B, it can be proven, according to [9], that the quadrotor dynamics are differentially flat. This means that there exists a smooth map, \mathbf{g} , such that the following is satisfied:

$$(\mathbf{r}, \mathbf{u}) = \mathbf{g}(\sigma, \dot{\sigma}, \ddot{\sigma}, \ddot{\sigma}, \ddot{\sigma}), \quad (5)$$

where $\mathbf{r} = [x, \phi, \theta, \psi, \dot{x}, p, q, r]^T$, with the angular rates being

$$[p, q, r]^T = \begin{bmatrix} c\theta & 0 & -c\phi s\theta \\ 0 & 1 & s\phi \\ s\theta & 0 & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}, \text{ where } c \text{ and } s \text{ represent}$$

cosine and sine respectively, and the input is defined as $\mathbf{u} = [f, M]^T$.

Following the derivations of [6] and [9], the explicit equations for the dynamic model will be derived below. The choice of flat outputs are defined as: $\sigma = [\vec{x}_w, \vec{y}_w, \vec{z}_w, \psi]^T$, where

$[\vec{x}_w, \vec{y}_w, \vec{z}_w]$ represent the coordinates of the inertial frame, and ψ represent the Euler angle, yaw. This is the angle, using Z-X-Y rotation sequence, with which \vec{z}_w is rotated. The Euler angles roll, ϕ , and pitch, θ , are defined similarly but with regards to rotation along \vec{x}_w , respectively, \vec{y}_w .

The rotation matrix, in terms of the Euler angles, is given by [6] as:

$${}^wR_b = \begin{bmatrix} c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi \\ c\theta s\psi + c\psi s\phi s\theta & c\phi c\psi & s\psi s\theta - c\psi c\theta s\phi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix}. \quad (6)$$

Rewriting equations (1) and (2) from Section II-B results in:

$$x = [x_w, y_w, z_w]^T = [\sigma_1, \sigma_2, \sigma_3]^T, \quad (7)$$

$$\dot{x} = [\dot{x}_w, \dot{y}_w, \dot{z}_w]^T = [\dot{\sigma}_1, \dot{\sigma}_2, \dot{\sigma}_3]^T \quad (8)$$

and

$$\ddot{x} = [\ddot{x}_w, \ddot{y}_w, \ddot{z}_w]^T = [\ddot{\sigma}_1, \ddot{\sigma}_2, \ddot{\sigma}_3]^T. \quad (9)$$

The rotation matrix, wR_b , from \mathcal{F}_w to \mathcal{F}_b , consists of the vectors of the body frame. The third component, z_b can be derived from (8), which results in the expression:

$$\hat{z}_b = \frac{\vec{t}}{\|\vec{t}\|}, \quad (10)$$

where,

$$\vec{t} = [\ddot{\sigma}_1, \ddot{\sigma}_2, \ddot{\sigma}_3 + g]^T. \quad (11)$$

The x-axis of an intermediate body frame, between \mathcal{F}_b and \mathcal{F}_w , is described by:

$$\vec{x}_c = [\cos\sigma_4, \sin\sigma_4, 0]^T. \quad (12)$$

Using the orthogonality principle between unit vectors, the following applies:

$$\hat{y}_b = \frac{\hat{z}_b \times \vec{x}_c}{\|\hat{z}_b \times \vec{x}_c\|}. \quad (13)$$

Assuming $\|\hat{z}_b \times \vec{x}_c\| \neq 0$, this yields:

$$\hat{x}_b = \hat{y}_b \times \hat{z}_b, \quad (14)$$

which results in the rotation matrix:

$${}^wR_b = [\vec{x}_b, \vec{y}_b, \vec{z}_b]. \quad (15)$$

Similarly, from (4), the angular velocity is expressed as:

$$\Omega = [\vec{x}_c \quad \vec{y}_b \quad \vec{z}_w] \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}. \quad (16)$$

D. Dynamics in terms of jerk

As mentioned in the introduction, the motion is affected by the jerks of each rotor. There are however two main aerodynamic forces that act on a quadrotor. There is the drag force that stem from the propellers and the drags when the UAV is flying. The latter is negligently small, which is why only the induced thrust of the former will be considered in this section [10]. Additionally, each rotor can be represented as a triple integrator that separately affect the generated motion. This relates to the paths' feasibility with respect to the rotors

capabilities, among other [11]. The jerk can be written as $j = \ddot{x} = (\ddot{x}_1, \ddot{x}_2, \ddot{x}_3)$. The input thrust f is computed by applying the Euclidean norm to (2), which is equivalent to:

$$f = \|\ddot{x} - \mathbf{g}\|. \quad (17)$$

Combining this derivative with (1) and (17) produces the following expressions:

$$j = \frac{1}{m}(R[\Omega]_{\times} f + R\dot{f}), \quad (18)$$

$$\dot{f} = m z_w^T R^{-1} j, \quad (19)$$

which can be rewritten with regard to the jerk, j , and thrust, f , being fixed components of the body rates, as:

$$\begin{bmatrix} \theta \\ -\phi \\ 0 \end{bmatrix} = \frac{1}{f} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} R^{-1} j, \quad (20)$$

where the yaw angle, ψ , is set to zero for convenience due to it not affecting the translational motion [12].

III. PROBLEM FORMULATION

The task is to find a feasible trajectory for a UAV traveling from an initial state X_o to a goal state X_g , in an environment filled with obstacles. The states consist of a position, velocity and acceleration. All of which are 3-dimensional vectors. In order to reach the desired results, the following goals should be achieved:

- 1) To create or to find an optimal trajectory from a time interval, τ , to a state space with regards to the dynamics of a quadrotor. The path should be generated such that it, from an initial state, reaches a goal state in an optimal way - meaning taking the shortest and most computationally efficient path.
- 2) To select or create a method for managing Collision detection. That is to say that the quadrotor is able to detect a collision at every given time instant.
- 3) To select or create a method for managing Obstacle avoidance, such that the obstacles can be avoided.
- 4) Use the given simulation environment and test several different scenarios to illustrate and validate the methods.

IV. MOTION PLANNING USING A*

In a proceeding thesis in 2020, Oscar Palfelt and Fredrik Skjernov created a motion planning algorithm, using A*. This section will summarize their findings, which later sections will be based on.

A. A*

A* is a pathfinding algorithm that is based on Dijkstra's Algorithm and Greedy Best-First Search. It is able to give as short of a path as the former algorithm, while reaching an equal computational cost to the latter. The computational cost is defined as the sum of the cost from the starting point to a vertex n , and the heuristic estimated cost from a vertex n to the goal. Using this, A* will compare the cost of different generated trajectories and only return the trajectory with the lowest cost [13].

The used graph is based on the flat outputs, see Section II-C. In order to find the optimal path, using A*, the subgraph U_f is found from Algorithm 1.

In [14], the trajectory is created using points chosen within a specific distance from the UAV, which can be described as the points on a sphere with a given radius. To make the algorithm work more efficiently, an angle and radius needs to be specified. This will generate a surface area, which will be checked for intermediate points [15]. For this purpose, Algorithm 1 is used from [15].

Algorithm 1: Finding subgraph U_f from graph U by building the tree of U .

Result: U_f

Create tree U with root $\sigma_{1,j}$;

Generate n trajectories $\sigma_{1,j} j \in 1, 2, \dots, n_t$;

Define distance to $\|X_g : d_g = \sigma_{i,j}(t = T)x_g\|$;

put $\sigma_{1,j}$ in queue ;

while $d_g \geq 0$ **do**

if *queue* \neq *empty* **then**

 Sample $X_{i,j} \forall j = \{1, 2, \dots, n\}$ from a spherical cap to obtain intermediate points;

 Generate motion primitive $\sigma_{i,j}$ for every $X_{i,j}$;

if $\sigma_{i,j}$ *collision free* **then**

 Create node $\sigma_{i,j}$ in U to store $\sigma_{i,j}$;

 Sort $\sigma_{i,j}$ into queue using A*;

else

 Attempt aggressive maneuver

else

 dequeue the new optimal node $\sigma_{i,j}$;

 calculate d_g with new node;

return No feasible trajectories

return U_f

B. Obstacle handling

The algorithm takes a number of points in which it will check for collisions. These points are used when selecting time indexes for which the position of the UAV will be extracted and controlled against the closest obstacles, both of which are modelled as polytopes. If an intersection occurs, the same procedure will be repeated but with the obstacle down-scaled in all directions to investigate the area around the obstacle, and then it will check for collisions again. If it does not occur any collisions at that stage, the edges of the obstacle are considered collision-free and aggressive maneuvers will be performed in that area. If it is not possible for the UAV to pass through the obstacles by applying aggressive maneuvers, a new path will try to be generated before deeming the path not-feasible.

Intersection between two polytopes is checked by extracting the vertices of the involved polytopes and checking if the distance between them is greater than the radius of the sphere [14], [15].

V. DEVELOPMENTS

A. Feasibility constraints

The input feasibility constraints are of interest to check for the true system's inputs f and Ω regarding the limits of specific

aspects, i.e. values that are not realistic to the quadrotor's capacity. For this case, input feasibility constraints for thrust and body rates are checked for every time interval τ and can be implemented as described in [8].

For the trajectory to be considered feasible, with respect to the thrust limits, it needs to fulfill the following conditions:

$$\max_{t \in \tau} f(t)^2 \leq f(t)_{\max}^2, \quad (21)$$

$$\min_{t \in \tau} f(t)^2 \leq f(t)_{\min}^2, \quad (22)$$

where f is defined in (17).

Input feasibility regarding body rates is also taken into account, where body rates magnitude can be described by an inequality of jerk and thrust, defined as:

$$\phi^2 + \theta^2 \leq \frac{1}{f^2} \|j\|^2. \quad (23)$$

If the interval τ is considered feasible, the input of the body rates must also fulfill the condition $\Omega^2 \leq \Omega_{\max}^2$, where Ω_{\max} is an experimental value that should represent a realistic angular velocity [8]. The position feasibility constraint needs to be implemented to test whether the UAV remains within the allowed space and does not crash into the defined floor. By describing the velocity of the quadrotor as a polynomial with the direction of the normal, the equation can be checked for the zero - meaning when the position is non-safe.

B. Obstacle handling

The original code checks for intersection by extracting the edges from the existing polytope and comparing them. This comparison is replaced by using functions looking for polytope intersection, which lowers the computational time.

This project models the geometric transformation of the UAV as a polytope, which is described by $P: \{x \in \mathbb{R}^3 : Ax \leq b\}$. Similarly, translated and rotated polytopes can be obtained by the following equations (24) - (25):

$$y \in P_2 : Ay \leq b, \quad (24)$$

$$z \in P_3 : Az \leq b, \quad (25)$$

where $y = x - o$, $z = R^T x$, with o as the translation vector and R as the rotation matrix.

Combining the equations results in the following expression for the translated and rotated polytope:

$$A'x \leq b', \quad (26)$$

where:

$$A' = AR^T, \quad (27)$$

$$b' = -AR^T o + b. \quad (28)$$

In order to make the code compatible with the new way of looking for intersections, it is necessary to redesign the collision checking. This is due to the fact that many functions are intertwined, which means that even small changes causes the program not to work.

The new collision avoidance is based on functions checking for polytope-intersection. This is done by selecting some time

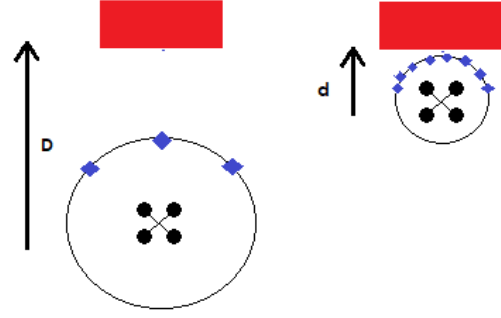


Fig. 2. The generation of waypoints depends on the distance to the obstacles. The red cubes represents the obstacles, the blue rhombus the waypoints and the black symbols the UAV. The distances between the obstacles and the UAVs are D , respectively d , where $d < D$.

interval, where the position of the UAV will be extracted at each time instant, and used to create a translated version of it that is used to check for intersection with all obstacles. If a collision occurs, the UAV is rotated and the intersection is controlled again. For the tested scenarios, the time interval is chosen to be between 0 and 4.5 seconds, where each trajectory is extracted during every 0.1 s.

C. Generate intermediate points

The A* algorithm generates intermediate points within a spherical cap, as explained in IV-A and in [16]. One problem is the fact that the parameters used in the previous thesis does not generate the same trajectory with the new collision avoidance. Therefore, this will also be changed. The new way of generating intermediate points is based on the distance between the UAV and the closest obstacle, which results in more variation in the parameters created. The same parameters will however still be used. These include the radius, r , of the sphere, the angular region, α , that specifies the region on the sphere where the waypoints n will be created within, and the motion primitive duration T . The relation between the parameters for choosing the points within a region of the sphere is described by:

$$\left| \frac{\arccos(\vec{p}_{i,j} \times \vec{v})}{r \|\vec{v}\|} \right| \leq \alpha, \quad (29)$$

where \vec{v} represent some vector belonging to \mathbb{R}^3 , and $\vec{p}_{i,j}$ is the vector from the origin to one of the intermediate points.

When the UAV reaches a distance d to the obstacles, more points will be generated in order to maximize the number of possible paths that the UAV can choose from. On the other hand, when the UAV is further away than distance d but closer than distance D to the obstacle, new parameters will be chosen that generates fewer intermediate points, but from a sphere with a larger radius. This can be seen in Figure 2. Furthermore, when the distance is greater than D , the UAV will look at an environment regarded as safe, and therefore search the path with $r \gg D$.

A pseudo-algorithm for this function is described in Algorithm 2. The algorithm is used for all the scenarios but with different parameters, see Section VI.

Algorithm 2: Choosing intermediate points from a sphere.

```

/* Algorithm that selects the best
   values of  $r, \alpha, T, n$  */
/*  $r$  = radius of sphere */
/*  $\alpha$  = angle between the end points
   of the sphere */
/*  $T$  = Time duration */
/*  $n$  = number of points on the sphere */
/*  $d$  and  $D$  are distances, defined as
    $d \leq D$ . */
/* min obstacle distance: the
   distance from the UAV to the
   closest obstacle. */
Result:  $n$  points on the sphere
if min obstacle distance  $\leq d$  then
    Choose parameters to create more, closely placed
    points  $n$  with shorter radius  $r$  on the sphere;
else
    if  $d \leq \text{min obstacle distance} \leq D$ , then
        Choose parameters to create fewer, sparsely
        placed points  $n$  on sphere;
    else
        Generate sparsely placed points, but with  $r$ 
        equal to the distance to the closest obstacle;
return  $r, \alpha, T, n$ 

```

VI. RESULTS

The extension to the code of the 2020's thesis report [15] was done in Python 3.7 and demonstrated visually in MATLAB. The program is available through Github [17]. There are four different scenarios, which include different types of red obstacles. The quadrotor was modeled as a blue, hexahedron polytope with the width 0.5 m, height 0.1 m, mass 1 kg and with an inertia matrix of $J = \begin{pmatrix} 0.082 & 0 & 0 \\ 0 & 0.0845 & 0 \\ 0 & 0 & 0.1377 \end{pmatrix}$ kgm².

The obstacles were, in each scenario, located between the initial and goal state. Note that these states are defined as vectors in the following order: position, velocity and acceleration.

For the input and position feasibility constraints, the parameters were chosen to be $f_{\min} = 5 \text{ m/s}^2$, $f_{\max} = 25 \text{ m/s}^2$ and $w_{\max} = 20 \text{ rad/s}$. The time during which the constraints are checked is defined to be $t_{\min} = 0.02 \text{ s}$. Additionally, the ground level was specified as $[0, 0, -5]$ and the direction as $[0, 0, 1]$ for the normal line.

The run time was taken as the mean of ten computations.

Scenario 1

This scenario tested the UAV's ability to find a path without the usage of aggressive maneuvers. The parameters can be seen in Table I and the result in Figure 3.

The program managed to generate a collision-free path from the initial position $X_0 = [[0, 0, 0][0, 0, 0][0, 0, 0]]$ to the final position $X_g = [[2, 2, 2][0, 0, 0][0, 0, 0]]$, while also fulfilling the input feasibility checks. The run time was 17.304 s compared

to the previous 163.9945 s, meaning the computational time became 89% faster.

TABLE I
VALUES OF THE PARAMETERS FOR SCENARIO 1.

Distance [m]	r [m]	α [rad]	T [s]	n
$2 \leq d_{obs}^1 \leq 10$	2	$\frac{11\pi}{36}$	1	2
$d_{obs} < 2$	1	$\frac{11\pi}{18}$	1	10
$d_{obs} \gg 10$	d_{obs}	$\frac{13\pi}{18}$	$\frac{r}{2}$	2

¹ d_{obs} refers to the distance from the UAV to the closest obstacle.

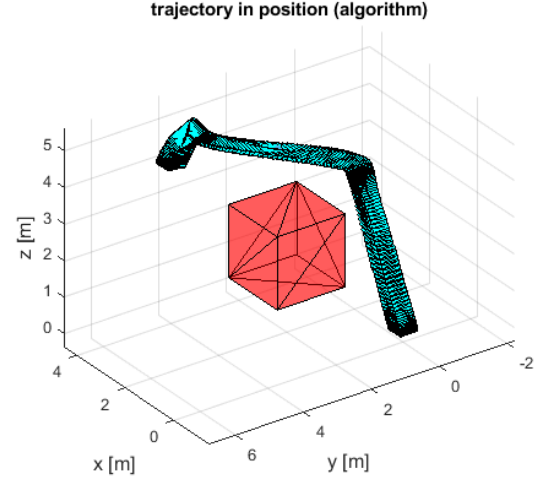


Fig. 3. The simulation environment for scenario 1.

Scenario 2

Scenario 2 is made up of five cubic obstacles placed closely together. As in scenario 1, the aim was to test the path planning algorithm, but in a more complex environment. The UAV did manage to travel from the initial state $X_g = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]$ to the goal state $X_g = [[4, 4, 4], [0, 0, 0], [0, 0, 0]]$ without colliding with the obstacles. The UAV fulfilled the constraints for input and positions feasibility, while travelling through a collision-free path, which can be seen in Figure 4. The parameters are shown in Table II.

The run time was found to be 460.67 s, in contrast to the previously measured result of 502.682 s.

TABLE II
VALUES OF THE PARAMETERS FOR SCENARIO 2.

Distance [m]	r [m]	α [rad]	T [s]	n
$2 \leq d_{obs} \leq 10$	2	$\frac{11\pi}{36}$	1	2
$d_{obs} < 2$	1	$\frac{11\pi}{18}$	1	10
$d_{obs} \gg 10$	d_{obs}	$\frac{13\pi}{18}$	$\frac{r}{2}$	2

Scenario 3

The purpose of scenario 3 is to test the aggressive maneuvers function, which is done with the use of two narrow gaps. The UAV did collide with both obstacles while travelling from the initial state $X_0 = [[0, -2, 0][0, 0, 0][0, 0, 0]]$ to the final state $X_g = [[0, 1.95, 0][0, 0, 0][0, 0, 0]]$, see Figure 5.

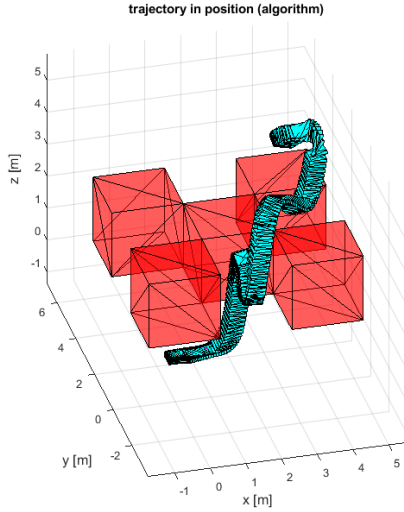


Fig. 4. The simulation environment for scenario 2.

Aggressive maneuvers were activated but did not appear in the simulations. The trajectory was however feasible with regard to the input and position constraints. The parameters are demonstrated in Table III.

The total iteration time for the simulation was measured to be 11.414 s vs. 234.344 s, showing a 95% faster computational time.

TABLE III
VALUES OF THE PARAMETERS FOR SCENARIO 3.

Distance [m]	r [m]	α [rad]	T [s]	n
$2 \leq d_{obs} \leq 8$	2	$\frac{\pi}{200}$	1	5
$d_{obs} < 2$	1	$\frac{29\pi}{200}$	1	10
$d_{obs} \gg 8$	d_{obs}	$\frac{13\pi}{18}$	$\frac{r}{2}$	2

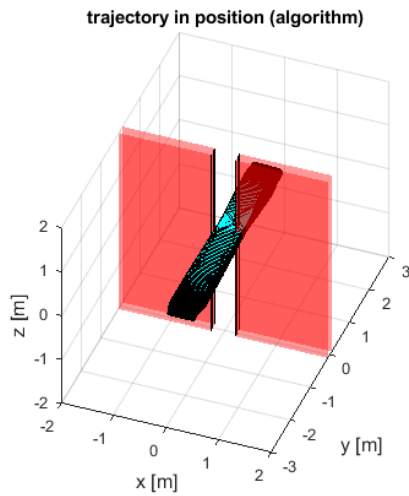


Fig. 5. The simulation environment for scenario 3.

Scenario 4

Scenario 4 is a new scenario, consisting of a clustered environment. The purpose of this scenario is to test all aspects

of the algorithm, as well as its capability to handle new scenarios. The parameters can be seen in Table IV.

The simulation resulted in the trajectory reaching the goal, and avoiding the obstacles in a compilation time of 455.3 s. The trajectory also fulfilled the input and position feasibility constraints. The result is illustrated in Figure 6.

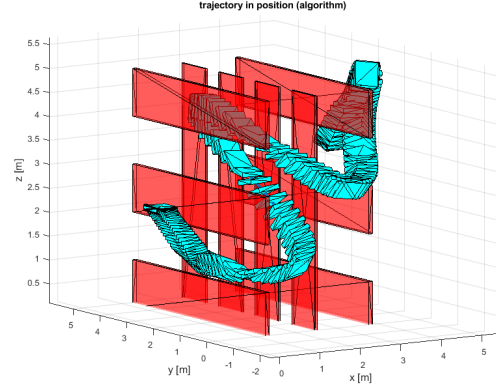


Fig. 6. The simulation environment for scenario 4.

TABLE IV
VALUES OF THE PARAMETERS FOR SCENARIO 4.

Distance [m]	r [m]	α [rad]	T [s]	n
$1 \leq d_{obs} \leq 10$	3	$\frac{\pi}{200}$	1	6
$d_{obs} < 1$	1	$\frac{100\pi}{18}$	1	10
$d_{obs} \gg 10$	d_{obs}	$\frac{11\pi}{18}$	$\frac{r}{2}$	2

TABLE V
THE COMPILATION TIME FOR THE SCENARIOS.

Scenario	New result [s]	Previous result (2020) [s]	Improvement [%]
1	17.304	163.995	89
2	460.67	502.68	8.3
3	11.414	234.344	95
4	455.286	-	-

VII. DISCUSSION

We compared our trajectories for scenario 1, 2 and 3, with those of the 2020 thesis report. We could see that the trajectories of scenarios 1 and 2 were able to find feasible paths from X_0 to X_g , while also having the feasibility constraints satisfied. In addition to that, the run time was improved for all scenarios, as seen in Table V. In scenario 3, the rotation of the UAV did not activate as it should have, which resulted in a collision.

We also implemented the new scenario 4. Our code managed to generate a feasible path and to satisfy the feasibility constraints. The code of the 2020 report could not generate a path for this scenario. Hence, it was not possible to compare the trajectories further.

Scenario 1 was implemented as a way of testing the collision avoidance, without letting the aggressive maneuver interfere with the result. The result was successful since the feasible path also had a 89% faster computational time, which is believed to be the result of using polytope-intersection. This

is supported by the reason that progress in the run time was made even after small parts of the code had been replaced with polytope-intersection.

In scenario 2, the computational time of our code was faster than the original, but it was not a significant difference. The major difference was the fact that the new trajectory was unnecessarily long. This indicates that a faster computation time does not necessarily result in a shorter path. The reason for this might be that not all obstacles are being detected accurately. This consequently means that not all the points are being registered for the pathfinding algorithm.

Furthermore, Algorithm 2 managed to find a path, while also avoiding obstacles. It is however important to note, that even though a good compilation time is necessary for autonomous decision-making - it is equally important to detect all environmental factors.

The comparison of scenario 3 showed a similar result as the former scenario did. It had a faster run time but an inadequate trajectory. The difference is that the new algorithm could not generate a collision-free path for this scenario. This supports the belief that the obstacles are not being detected accurately, possibly being a consequence of the obstacles' dimensions not being included properly in the polytope-conversion.

It did not make a difference whether or not aggressive maneuvers were activated, the result was still a path that intersected with the obstacles. In contrast to this, the previous code did manage to find a collision-free path when aggressive maneuvers were activated. One possible reason behind this, could be due to the design of the original code. It was noted that many sensitive parameters were occurring, which causes the program to be very specific and not adaptable to other scenarios. Other problems arise when making small variations to the code, such as the positions, which resulted in the code not finding a feasible path. To solve these issues, it could be necessary to redesign the entire code and thereby make it adaptable to different kinds of scenarios.

Another problem is how the program decides to activate rotations. This part is currently not behaving as expected, which leads to a lack of necessary rotations.

Scenario 4 is a newly implemented scenario that was not considered in the 2020 report. We tested their code with the scenario, but it could not generate a path. Despite our being successful, the trajectory did not move as expected. Instead of moving linearly, it travels longer and avoiding most obstacles by traveling outside the obstacle-filled area. This, while activating aggressive maneuvers to travel up and down. This seems to be a consequence of rotations not being activated correctly. From Figure 6, it is clear that the UAV never actually rotates. It does however seem to detect the obstacles, since it travels among them when trying to reposition itself.

One of the major challenges we encountered throughout the project was to implement new methods into the previous code. The code was not adaptive to other situations, which consequently meant more time was spent on handling the code, instead of improving methods and algorithms.

A. Improvements

There are many possible improvements that can be made to 2020 code that we have yet to do. One necessary development

would be for the code to always find a feasible path whenever one exists. Another improvement would be to make the program generate its own values, or to completely remove the dependence on the parameters.

VIII. CONCLUSION

This report investigated how to improve an already existing path planning algorithm. Some of the changes include using built in functions intended to find polytope-intersection, as well as basing the collision avoidance algorithm on these whenever possible.

Trajectories were generated for four different scenarios, three of which could be used to compare the code of [14] with the extended version from [17]. The previous code generated shorter paths, while the new code generated trajectories faster. In addition to that, the extended version always generated feasible paths, with respect to input and position. Although, for one of the three scenarios, it could not generate a collision free-path, due to the rotations not being activated correctly. For the other two scenarios, the trajectories were collision-free.

Other than that, the new code could successfully generate a collision-free and feasible path for a new scenario, whereas the previous code failed to do so.

Improvements need to be done for the collision detection. Not all obstacles are identified correctly, which makes the extended algorithm very flawed. It is also necessary to improve the functions handling rotations, which currently are not working as expected.

ACKNOWLEDGMENT

We would like to express our deepest gratitude to our supervisor, Xiao Tan, for guiding and supporting us throughout the project.

REFERENCES

- [1] Insider. (2021, Jan.) Drone technology uses and applications for commercial, industrial and military drones in 2021 and the future. [Online]. Available: <https://www.businessinsider.com/drone-technology-uses-applications?r=US&IR=>
- [2] Transportstyrelsen. (2021, Nov.) Drones – Unmanned aircraft. [Online]. Available: <https://www.transportstyrelsen.se/en/aviation/Aircraft/drones--unmanned-aircraft/>
- [3] TWI. (2022) What is an autonomous vehicle?. [Online]. Available: <https://www.twi-global.com/technical-knowledge/faqs/what-is-an-autonomous-vehicle>
- [4] A. Montazeri, A. Can, and I. H. Imran, "Chapter 3 - Unmanned aerial systems: autonomy, cognition, and control," in *Unmanned Aerial Systems*, ser. Advances in Nonlinear Dynamics and Chaos (ANDC), A. Koubaa and A. T. Azar, Eds. Academic Press, 2021, pp. 47–80 [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780128202760000108>
- [5] A. S. Matveev, A. V. Savkin, M. Hoy, and C. Wang, "3 - Survey of algorithms for safe navigation of mobile robots in complex environments," in *Safe Robot Navigation Among Moving and Steady Obstacles*, A. S. Matveev, A. V. Savkin, M. Hoy, and C. Wang, Eds. Butterworth-Heinemann, 2016, pp. 21–49 [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780128037300000032>
- [6] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robotics and Automation magazine*, vol. 19, no. 3, pp. 20–32, 2012.
- [7] B. Penin, P. Giordano, and F. Chaumette, "Vision-based reactive planning for aggressive target tracking while avoiding collisions and occlusions," *IEEE Robotics and Automation Letters*, vol. PP, 07 2018.

- [8] M. W. Mueller, M. Hehn, and R. D'Andrea, "A computationally efficient motion primitive for quadcopter trajectory generation," *IEEE transactions on robotics*, vol. 31, no. 6, pp. 1294–1310, 2015.
- [9] D. Mellinger, *Trajectory generation and control for quadrotors*. University of Pennsylvania, 2012.
- [10] J. Medrano, F. Yumbla, S. Jeong, I. Choi, Y. Park, E. Auh, and H. Moon, "Jerk estimation for quadrotor based on differential flatness," in *2020 17th International Conference on Ubiquitous Robots (UR)*. IEEE, 2020, pp. 99–104.
- [11] S. Ravindran, V. T. Minh, J. Pumwa. (2014) Feasible path planning for autonomous vehicles. [Online]. Available: <https://www.hindawi.com/journals/mpe/2014/317494/>
- [12] M. Hehn and R. D'Andrea, "Quadcopter trajectory generation and control," *IFAC proceedings Volumes*, vol. 44, no. 1, pp. 1485–1491, 2011.
- [13] A. Patel. (2022, Apr.) Introduction to A*. [Online]. Available: <http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html>
- [14] O. Palfelt and F. Skjernov. (2020, May) Code for bachelor thesis 'Motion planning for aggressive flights of an unmanned aerial vehicle'. [Online]. Available: <https://gits-15.sys.kth.se/palfelt/KEX>
- [15] O. Palfelt and F. Skjernov, "Motion planning for aggressive flights of an unmanned aerial vehicle," *Bsc. thesis, KTH Stockholm*, 2020.
- [16] V. Rospotniuk and R. Small, "Optimal any-angle pathfinding on a sphere," *arXiv preprint arXiv:2004.12781*, 2020.
- [17] F. Femic and C. Smith. (2022, Apr.) Code for bachelor thesis 'Motion planning for aggressive flights of an unmanned aerial vehicle'. [Online]. Available: <https://github.com/corsmi/kex2022>

Collaborative Control of Autonomous Ground Vehicles

Gustav Thorén and Moa Säll

Abstract—Autonomous ground vehicles (AGVs) is a growing field within research. AGVs are used in areas like reconnaissance, surveillance, transportation and self-driving cars. The goal of this project is to drive a system of five AGVs modelled as differential-drive vehicles along an arbitrary path through a field of obstacles while holding a formation. The goal is achieved by dividing the project into three subprojects. The first subproject is trajectory tracking of one AGV. This is achieved by using the differential-drive model and driving the tracking error of the system to zero. The second subproject is formation control, where a displacement based, double integrator model is used to get five AGVs to hold a formation of an equilateral triangle while following a path. The third subproject is collision avoidance between AGVs and static obstacles placed along the predetermined path. Collision avoidance is achieved by adding a repulsive potential field around the AGVs and obstacles. All three subprojects are then combined to achieve the goal of the project. Finally, simulations are done in Matlab which confirms that the proposed models are correct.

Sammanfattning—Autonoma vägfordon är ett växande område inom forskning. Autonoma vägfordon används inom områden som spaning, övervakning, transporter och självkörande bilar. Målet med det här projektet är att köra ett system med fem autonoma vägfordon modellerade som differentialdrivna fordon längsmed en slumpmässig väg genom ett fält med hinder samtidigt som de håller en formation. Målet uppnås genom att dela upp projektet i tre delprojekt. Det första delprojektet är banspårning med ett autonomt vägfordon. Det görs genom att använda den differentialdrivna modellen och driva systemets spårningsfel till noll. Det andra delprojektet är formationshållning där en förskjutningsbaserad dubbelintegratormodell används för att få fem fordon att följa en väg samtidigt som de håller formen av en liksidig triangel. Det tredje delprojektet handlar om att undvika kollision mellan fordonen och statiska hinder som placerats på vägen. Kollisionsundvikning uppnås genom att lägga på ett repellerande potentialfält runt alla agenter och hinder. Alla tre delprojekt kombineras sedan för att lösa projektmålet. Slutligen görs simuleringar i Matlab vilket bekräftar att de framtagna modellerna är korrekta.

Index Terms—autonomous ground vehicles, trajectory tracking, formation control, collision avoidance, repulsion field

Supervisors: Fei Chen

TRITA number: TRITA-EECS-EX-2022:125

I. INTRODUCTION

According to [1] the number of published research papers on Autonomous Ground Vehicles (AGVs) have increased a lot since 2013, it is clear that it is a growing field within research. AGVs are used for reconnaissance and surveillance [2], transportation [3], and self-driving cars [4] for both civilians and the military [3]. Adding efficient and safe self-driving cars to everyday traffic will lead to safer and more

efficient transportation. The increased efficiency will also lead to less environmental pollution from inefficient driving [4]. It is obvious that the applications are plentiful and important, and it is therefore essential to thoroughly study how to control AGVs. In this project, the focus is on controlling multiple AGVs simultaneously working together.

In [5] and [6], it is shown that a formation of three to four differential-drive vehicles can follow a straight path while avoiding collision with obstacles and with each other. In [5], a formation of four differential-drive vehicles permanently deform their formation in order to pass between two obstacles. The obstacles in [6] are also placed a large distance from each other, so that the whole formation of vehicles can fit between them without trouble. The report also mentions how it would be interesting to combine path following, formation control and obstacle avoidance into one. The goal of this project is to achieve this while also adding more differential-drive vehicles, follow a more complicated reference path, and use a larger set of obstacles placed on the desired path.

II. PRELIMINARIES

A. Notations

There are some mathematical notations that occur in this paper. The matrices are written in bold, capital letters like \mathbf{M} . A letter with a bar notation like \bar{v} , is used to describe a vector. A T denote that it is the transpose of a vector or a matrix. In that case, it will appear like \bar{v}^T or \mathbf{M}^T . A dot over a variable imply that it is a time derivative, that is, \dot{x} is the derivative of x with respect to t . Skewed letters, like k , means that it is a scalar.

B. Differential-Drive Vehicle

A differential-drive vehicle has been used in this project to simulate the ground vehicles, since it is the simplest model for simulating an AGV. This model simulates the AGVs with only one pair of wheels, each independently controlled from the other. In a coordinate system the position of the vehicle is described by the coordinates x , y and θ , where the first two are the place with respect to the x - and y -axis and θ is the angle that the vehicle is directed, an example is shown in Fig. 1. A two wheeled model is used since the dynamics does not significantly differ from models with more wheels, while it is easier to model and control. [7]. In this paper, the number of differential-drive vehicles used is denoted with N .

III. PROBLEM DESCRIPTION

The goal of this project is to study cooperative control of AGVs by studying trajectory tracking, formation control and

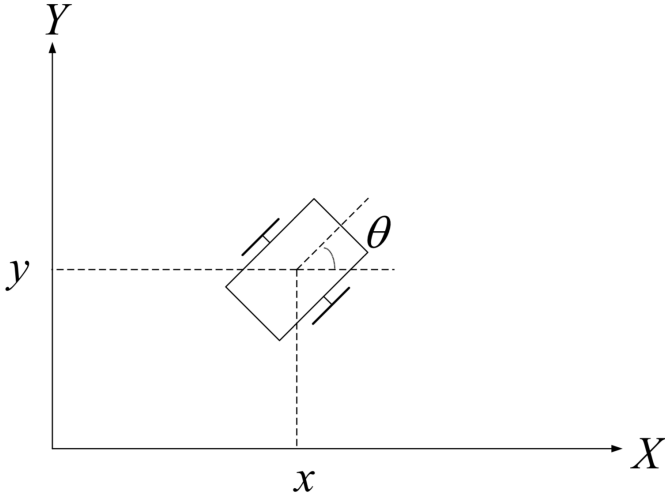


Fig. 1. An example of the differential-drive vehicle model from [7].

collision avoidance. The main goal is to have a formation of five differential-drive vehicles following an arbitrary path through a field of obstacles, after passing through the field the differential-drive vehicles should also restore their initial formation and velocity. In order to achieve this goal, the project is split into three subprojects which build upon each other. Each subproject has their own goal which are designed so that when all subprojects are achieved, the main goal is also achieved.

A. Trajectory Tracking

The first subproject is trajectory tracking, and the goal is to have a differential-drive vehicle follow a simple, smooth, and nonlinear two-dimensional reference path at a constant speed. Regardless of the starting position and velocity, the differential-drive vehicle should find and follow the path and velocity asymptotically.

B. Formation Control

The second subproject is formation control, and the goal is to have five differential-drive vehicles with random starting positions reach the formation of an equilateral triangle. They are also supposed to hold that formation while following a path and velocity defined the same way as the first subproject.

C. Collision Avoidance

The third and last subproject is collision avoidance, where the differential-drive vehicles should avoid collision with each other and a few static obstacles placed on the path. After avoiding the obstacles, the differential-drive vehicles should revert to a stable formation the same way as defined by the first and second subprojects.

IV. TRAJECTORY TRACKING

In this project, the kinematic model used for the differential-drive vehicle (henceforth referred to as agent), is a nonlinear

and nonholonomic Cartesian model

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

where x and y is the position, θ is the angle, u_1 is the speed and u_2 is the angular velocity. A nonlinear model is often difficult to control compared to a linear model. In order to get around this issue and make it easier to control the model described in (1), it is transformed into chained form. The transformation from Cartesian form into chained form is

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \theta \\ x \sin(\theta) - y \cos(\theta) \\ x \cos(\theta) + y \sin(\theta) \end{bmatrix}, \quad (2)$$

where x , y and θ is the state of the agent in Cartesian form. Chained form shares many similarities with a linear system and is therefore easier to control and design control systems for compared to the Cartesian form[8].

The trajectory tracking is in this project achieved by controlling the error of the system. When the error of the system converges to zero, the system converges to the desired state. The error of the model in chained form is calculated as

$$\bar{x}_e = [x_{1e} \quad x_{2e} \quad x_{3e}]^T \triangleq \bar{x} - \bar{x}_d,$$

where \bar{x} is the state in chained form and \bar{x}_d is the desired state of the agent in chained form. It can be shown from (1) and (2) that if the error in the chained form converges to zero, the error in the Cartesian form also converges to zero. As such, it is sufficient to drive the error in chained form to zero in order to control the agent as desired.

In this project, the model (2) is driven to achieve trajectory tracking with the asymptotically stable system proposed by [7]. The proposed system is described by the equations

$$\dot{x}_{1e} = v_1 \quad (3)$$

$$\dot{\bar{z}} = w_{1,d}(t)\mathbf{A}_c\bar{z} + \mathbf{B}_cv_2 + \mathbf{G}v_1 \quad (4)$$

where $\bar{z} \triangleq [x_{2e} \quad x_{3e}]^T$, v_1 and v_2 are control variables, $\mathbf{A}_c = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $\mathbf{B}_c = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $\mathbf{G} = \begin{bmatrix} x_{3e} + x_{3d} \\ 0 \end{bmatrix}$ and $w_{1,d}$ is the desired control input of the chained form which is equivalent to the desired angular velocity in the nonholonomic model (1).

The system described by (3) is a simple system which can easily be solved by defining v_1 to be negatively proportional to x_{1e} . The used solution is the solution proposed by [7],

$$v_1 = -x_{1e}\sqrt{\frac{q}{r_1}},$$

where q and r_1 are constants.

The system described by equation (4) is built up of two systems added together, $\dot{\bar{z}} = w_{1,d}(t)\mathbf{A}_c\bar{z} + \mathbf{B}_cv_2$ and $\dot{\bar{z}} = \mathbf{G}v_1$. The second system ensures coupling between (3) and (4). Since v_1 is designed to stabilise towards zero, this system will have no impact on the stability of the system. The first system is in a controllable canonical form when \mathbf{A}_c and \mathbf{B}_c are chosen as above. For the error to converge to zero, it is sufficient for v_2 to be negatively proportional to \bar{z} . Designing v_2 in a smart way can significantly reduce the oscillations and

time it takes for the error to converge to zero. The value used is the value proposed by [7],

$$v_2 = -\mathbf{B}_c^T \mathbf{P}_2(t) \frac{\bar{z}}{r_2}. \quad (5)$$

where $\mathbf{P}_2(t)$ is the solution to the differential Riccati equation

$$\dot{\mathbf{P}}_2 = \mathbf{P}_2 \mathbf{A}_c w_{1,d}(t) + w_{1,d}(t) \mathbf{A}_c^T \mathbf{P}_2 - \mathbf{P}_2 \mathbf{B}_c \mathbf{B}_c^T \frac{\mathbf{P}_2}{r_2} + \mathbf{Q} \quad (6)$$

where r_2 and \mathbf{Q} are constants. The error will converge to zero when $\mathbf{P}_2(t)$ is positive definite, which according to [7] is guaranteed when \mathbf{Q} is positive definite. Using (6) when designing v_2 , ensures that the model will respond both to the error and to the desired path, which reduces oscillations and settling time of the model.

V. FORMATION CONTROL

As described by [9] there are three main models to implement formation control: distance-based, position-based, and displacement-based formation control. Position-based formation control tries to achieve a formation by driving all the agents to a determined position in a global coordinate system. It is expensive, since it requires all the agents to be capable of communicating and determining their position precisely over large distances. On the other hand, this means that the agents don't need the capability to communicate with each other to achieve a formation.

Distance-based formation control requires no coordinate system, and instead tries to hold the formation by driving the agents to hold a specific distance from each other. The main problem with this model is that to achieve a rigid formation, most of the agents need to constantly communicate with each other. Even then, it will always be possible to stabilise the formation in the mirror-symmetrical position, which might not always be ideal.

The model used for this project is the displacement-based model, which is a combination of the other two. The displacement-based model drives the model by aligning all the agents to the desired formation in a local coordinate system and then aligning this coordinate system to a global coordinate system.

When exploring models for formation control, it makes sense to consider the family of double integrator models, since it is possible to easily transform them into the nonholonomic model (1). A true double integrator model for formation control can be described as

$$\begin{cases} \dot{\bar{p}}_i = \bar{v}_i \\ \dot{\bar{v}}_i = \bar{u}_i \end{cases} \quad (7)$$

for agents $i = 1, 2, 3, \dots, N$, where $\bar{p}_i \in \mathbb{R}^2$ is the position, $\bar{v}_i \in \mathbb{R}^2$ is the velocity and $\bar{u}_i \in \mathbb{R}^2$ is the control input of the agent i . The model proposed by [9] is a true double integrator model described by the equation

$$\begin{bmatrix} \dot{\bar{p}}_p \\ \dot{\bar{v}}_v \end{bmatrix} = \begin{bmatrix} 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & \mathbf{I} \\ -k\mathbf{L} & 0 & -k\mathbf{L} & 0 \\ 0 & -k\mathbf{L} & 0 & -k\mathbf{L} \end{bmatrix} \begin{bmatrix} \bar{e}_p \\ \bar{e}_v \end{bmatrix} \quad (8)$$

where k is a positive control constant, $\bar{e}_p \in \mathbb{R}^{2N}$ denotes the error in the position, $\bar{e}_v \in \mathbb{R}^{2N}$ denotes the error in the velocity for a 2-dimensional case and $\mathbf{L} \in \mathbb{R}^{N \times N}$ is the graph Laplacian of the agents. It is shown how (8) drives the formation to the desired state if and only if the matrix has exactly N zero eigenvalues and all non-zero eigenvalues have negative real parts.

The model used in this project is a modification of the proposed displacement-based double integrator model (8). The model is described by the equation

$$\begin{bmatrix} \dot{\bar{p}} \\ \dot{\bar{v}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & \mathbf{I} \\ -\mathbf{L} & 0 & -\mathbf{L} & 0 \\ 0 & -\mathbf{L} & 0 & -\mathbf{L} \end{bmatrix} \begin{bmatrix} \bar{p} \\ \bar{v} \end{bmatrix} + \begin{bmatrix} \dot{\bar{p}}_d \\ \bar{p}_{rel} \end{bmatrix}, \quad (9)$$

where $\bar{p} \in \mathbb{R}^{2N}$ is the position, $\bar{v} \in \mathbb{R}^{2N}$ is the velocity, $\dot{\bar{p}}_d \in \mathbb{R}^{2N}$ is the desired velocity for the formation and $\bar{p}_{rel} \in \mathbb{R}^{2N}$ is the desired relative position. The model (9) is essentially two models added together. The first model is a double integrator model driven by \bar{p}_{rel} , which controls the displacement of all the agents so that the desired formation can be achieved. The second model is a single integrator model driven by $\dot{\bar{p}}_d$, which drives the entire formation in 2-dimensional space as required.

There are two main advantages to using model (9) over model (8). The first advantage is that model (9) does not require transforming the error back into actual driving output, which allows for simpler and faster simulations. The second advantage is that model (9) allows for easier manipulation of the velocity using the single integrator model. By utilising the single integrator model there is no need to approximate the desired acceleration to achieve the desired velocity, which would be required with a true double integrator model, instead, the desired velocity can be achieved by direct input into the system.

VI. COLLISION AVOIDANCE

Collision avoidance is enforced by adding a repulsive potential field to all the agents and static obstacles placed on the field. A static obstacle is defined as an obstacle whose position is not influenced by external factors. In this project, stationary static obstacles were used. The obstacles are handled as stationary agents which are not part of the incidence matrix for the formation control and therefore stay stationary.

The model chosen for the repulsive potential field is the one described by [10] which requires the repulsive field between agents i and j , $V_{ij} : \mathbb{R}^2 \rightarrow \mathbb{R}_+$, to possess five properties:

- 1) V_{ij} has to be a function of the square of the distance between agents i and j such that

$$V_{ij} = V_{ij} \left(\underbrace{\|p_i - p_j\|^2}_{\beta_{ij}} \right) = V_{ij}(\beta_{ij}) \quad (10)$$

where $p_i \in \mathbb{R}^2$ and $p_j \in \mathbb{R}^2$ is the position of agent i and j .

- 2) The maximum value of V_{ij} must be found where $\beta_{ij} \rightarrow 0$. For an unbounded potential $V_{ij} \rightarrow \infty$ when $\beta_{ij} \rightarrow 0$.
- 3) V_{ij} is continuously differentiable in all of \mathbb{R}^2 .

- 4) When $\beta_{ij} > d^2$ where d is the edge of the potential field $\frac{\partial V_{ij}}{\partial p_i}$ and V_{ij} must be zero.
- 5) The partial derivative $\rho_{ij} \triangleq \frac{\partial V_{ij}}{\partial \beta_{ij}}$ must satisfy $\rho_{ij} = 0$ when $\beta_{ij} \geq d^2$ and $\rho_{ij} < 0$ when $0 < \beta_{ij} < d^2$.

For this project V_{ij} is designed as an unbounded repulsive potential field such that $V_{ij} \rightarrow \infty$ when $\beta_{ij} \rightarrow 0$. This guarantees that agents never collides with each other or static objects.

Repulsion field dynamics are handled by extending the formation control model (9) with an extra term,

$$\begin{bmatrix} \dot{\bar{p}} \\ \dot{\bar{v}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & \mathbf{I} \\ -\mathbf{L} & 0 & -\mathbf{L} & 0 \\ 0 & -\mathbf{L} & 0 & -\mathbf{L} \end{bmatrix} \begin{bmatrix} \bar{p} \\ \bar{v} \end{bmatrix} + \begin{bmatrix} \dot{\bar{p}}_d \\ \dot{\bar{p}}_{rel} \end{bmatrix} + \begin{bmatrix} \bar{p}_{rep} \\ 0 \end{bmatrix} \quad (11)$$

where $\bar{p}_{rep} \in \mathbb{R}^{2N}$ is the repulsive force between the agents and obstacles.

The repulsive force \bar{p}_{rep} is calculated from the repulsion field matrix \mathbf{R} . The matrix \mathbf{R} is designed as an extension of the model proposed by [11]. The proposed model is, for a system where all agents can sense each other, a symmetric matrix defined by

$$\mathbf{R}_{ij} = \begin{cases} -\rho_{ij}, & i \neq j \\ \sum_{j \neq i} \rho_{ij}, & i = j \end{cases} \quad (12)$$

where ρ_{ij} is the partial derivative of (10) with respect to β_{ij} . In this project this matrix is extended with one column for every static obstacle. Extending the matrix this way ensures a one-sided repulsive force from the obstacles to the agents. It is not strictly necessary that all agents can sense each other at all times, it is sufficient that they can sense all neighbouring agents when $\beta_{ij} < d$. For this project, the agents are modelled to be capable of perfectly sensing all other agents and obstacles in order to simplify the simulation.

VII. SIMULATION

The simulations were done in Matlab for all three subprojects.

A. Trajectory Tracking

The trajectory tracking is simulated by running model (3) and (4) with the parameters listed in Table I. Model (3) and (4) operates on chained form, so it is first necessary to transform \bar{p} and \bar{p}_d into chained form using (2). The chained form drive parameter $w_{1,d}(t)$, which is required for (4) and (5), is equivalent to the Cartesian form drive parameter $u_{2,d}(t)$.

After running the model, the result is given in chained form error and needs to be transformed back into Cartesian form position. The chained form position is derived from the chain form error as $\bar{x} = \bar{x}_d + \bar{x}_e$. The chained form position is then transformed into the Cartesian form position using

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} x_3 \cos(x_1) + x_2 \sin(x_1) \\ x_3 \sin(x_1) - x_2 \cos(x_1) \\ x_1 \end{bmatrix}.$$

Fig. 2 and Fig. 3 shows the result after running the simulation in Cartesian form. The actual and the desired trajectory of

TABLE I
PARAMETERS FOR SIMULATING TRAJECTORY TRACKING

Parameter and value	Explanation	Unit
$r_1 = 1$	Scalar constant	-
$r_2 = 1$	Scalar constant	-
$q = 10$	Scalar constant	-
$\mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	Positive definite matrix for calculating \mathbf{P}_2	-
$T = 30$	Time simulated	s
$\mathbf{P}_2(0) = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$	Start value for \mathbf{P}_2	-
$\bar{p} = \begin{bmatrix} 0 & 0 & \frac{\pi}{2} \end{bmatrix}^T$	Start position for the agent given in x, y and θ	[m, m, rad]
$\bar{p}_d = \begin{bmatrix} 0 & 5 & \frac{\pi}{4} \end{bmatrix}^T$	Desired start position given in x, y and θ	[m, m, rad]
$u_{1,d} = 2$	Desired speed of the agent	m/s
$u_{2,d}(t) = \begin{cases} \frac{\pi}{10}, & t \leq 5 \\ -\frac{\pi}{10}, & 5 < t \leq 15 \\ \cos(t + \frac{\pi}{4.4}), & 15 < t \end{cases}$	Desired angular velocity of the agent	rad/s

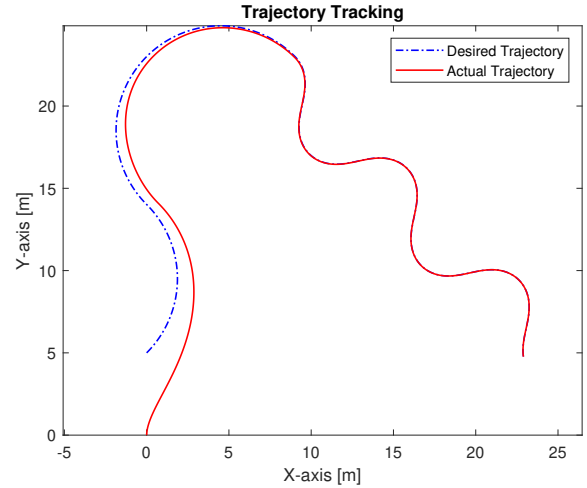
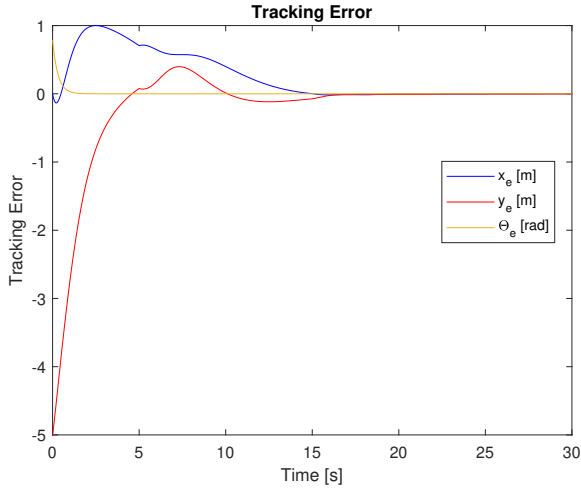


Fig. 2. Compares the actual and the desired trajectory of the agent for the simulation of trajectory tracking.

the agent is shown in Fig. 2. Fig. 3 shows the Cartesian form error during the simulation. The figures shows how the error converges to zero and the desired path is found and followed. That the error is zero also implies that the actual velocity is equivalent to the desired velocity.

B. Formation Control

In this project, the goal for Formation Control is for five agents to track a path while holding a formation of an

Fig. 3. Error in x, y and θ during the simulation of trajectory tracking.TABLE II
INITIAL COORDINATES FOR THE AGENTS

Agent number	Initial placement (x,y)
1	(-8,6)
2	(-6,-6)
3	(0.5,-3)
4	(5,-13)
5	(-12,0)

equilateral triangle, see Fig. 4. This is achieved by connecting the five agents in a graph and using the model (9) to control the formation. The chosen graph can be seen in Fig. 4 and is represented by the incidence matrix

$$\mathbf{D} = \begin{bmatrix} -1 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \end{bmatrix}, \quad (13)$$

which can be used to derive the graph Laplacian $\mathbf{L} = \mathbf{D}\mathbf{D}^T$ [12].

Model (9) is driven by $\dot{\bar{p}}_d$ and \bar{p}_{rel} where $\dot{\bar{p}}_d$ drives the movement and \bar{p}_{rel} drives the formation control independently of each other. There are multiple options for determining these driving variables. In this project, $\dot{\bar{p}}_d$ is chosen as the movement gained as a result from driving the mean of the starting position for the agents as if it was a single, nonholonomic agent. The method used for this is the same as in the first subproject described by equations (3) and (4). This is done to ensure that the movement of the formation is possible for a nonholonomic agent to follow. This simplifies the transform from the double integrator model into the nonholonomic model.

The side of the equilateral triangle is chosen to be eight meters, which correspond to the displacement along the edges as shown in Table III. In order to make the arrow point towards the trajectory, this displacement is turned towards the desired

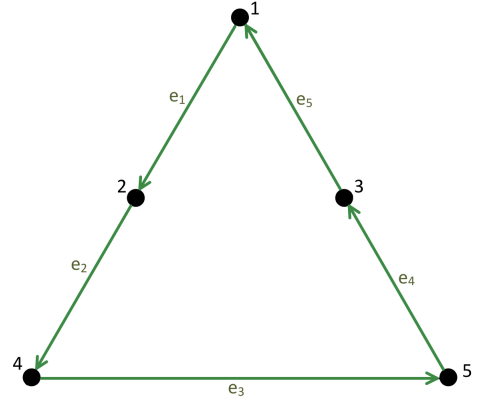


Fig. 4. The desired formation with five agents.

TABLE III
RELATIVE DISPLACEMENT ALONG THE EDGES OF THE GRAPH

Edge number	Relative displacement (x_{rel}, y_{rel})
1	$(-2\sqrt{3}, 2)$
2	$(-2\sqrt{3}, 2)$
3	$(0, -8)$
4	$(2\sqrt{3}, 2)$
5	$(2\sqrt{3}, 2)$

angle, α , using

$$\begin{cases} x_{T,i} = x_{rel,i}\cos(\alpha) - y_{rel,i}\sin(\alpha) \\ y_{T,i} = x_{rel,i}\sin(\alpha) + y_{rel,i}\cos(\alpha) \end{cases} \quad (14)$$

where $x_{T,i}$ and $y_{T,i}$ is the desired displacement along edge i taken from Table III. In this project, α is chosen by using the same angle as when calculating $\dot{\bar{p}}_d$. An alternative is calculating α from the current trajectory of the formation, which might be a better alternative when the system can not be perfectly modelled, such as when applying the model to physical agents. Finally, \bar{p}_{rel} is calculated by transforming the edge displacement into position displacement,

$$\dot{\bar{p}}_{rel} = \begin{bmatrix} \mathbf{D}\bar{x}_T \\ \mathbf{D}\bar{y}_T \end{bmatrix},$$

where \bar{x}_T and \bar{y}_T is the five element column vector from (14).

The results from the simulation are shown in Fig. 5, 6 and 7. Fig. 5 shows the formation control without movement. Fig. 6 shows the simulation when following a predetermined path with the starting positions as in Table II. Fig. 7 shows how the distance error converges to zero in roughly ten seconds and then stabilises.

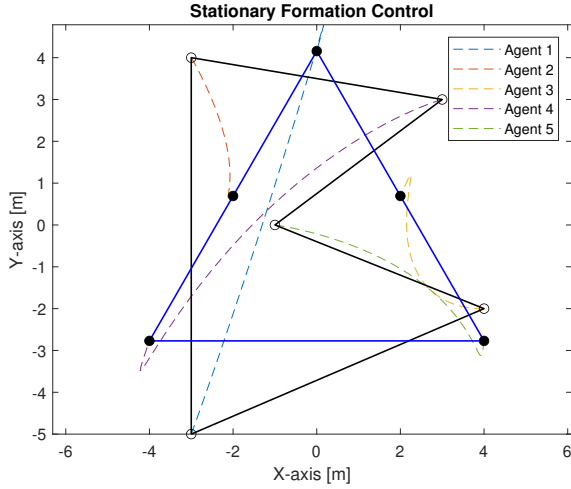


Fig. 5. Stationary formation control of five agents. The agents are moving into the desired formation and then staying stationary.

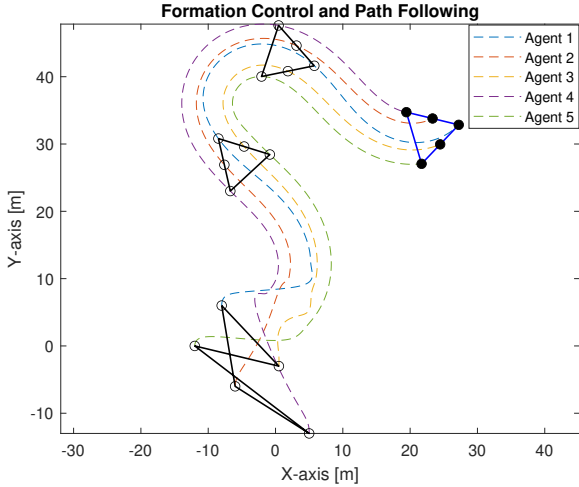


Fig. 6. Formation control with five agents following a predetermined path. The agents are holding a formation which points towards and follows the determined path.

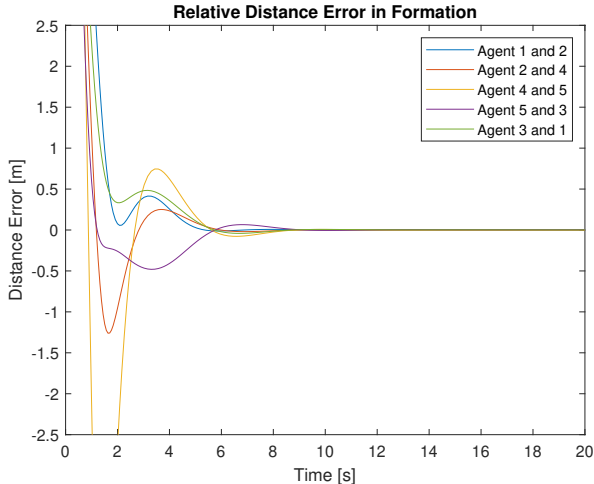


Fig. 7. Distance error for the formation following a predetermined path.

TABLE IV
CONSTANTS FOR SIMULATION OF COLLISION AVOIDANCE

Parameter	Value
d^2	$\frac{9}{e^1}$
c	$\frac{3}{e^1}$
h	$\frac{1}{4}$
p	$\frac{36}{e^2}$

TABLE V
POSITIONS OF STATIONARY OBSTACLES

Obstacle	Position (x,y)	Obstacle	Position (x,y)
1	(-3, 35)	7	(0, 38)
2	(-2, 35)	8	(0, 39)
3	(-1, 35)	9	(0, 40)
4	(0, 35)	10	(0, 41)
5	(0, 36)	11	(0, 42)
6	(0, 37)	12	(0, 43)

C. Collision Avoidance

When simulating collision avoidance, the repulsive potential field (10) is chosen to be

$$V_{ij}(\beta_{ij}) = \begin{cases} p \ln\left(\frac{1}{\beta_{ij}}\right), & \beta_{ij} < c \\ h(\beta_{ij} - d^2)^2, & c \leq \beta_{ij} < d^2 \\ 0, & \beta_{ij} \geq d^2 \end{cases}$$

which gives the partial derivative as

$$\rho_{ij}(\beta_{ij}) = \begin{cases} \frac{p}{\beta_{ij}}, & \beta_{ij} < c \\ 2h(\beta_{ij} - d^2), & c \leq \beta_{ij} < d^2 \\ 0, & \beta_{ij} \geq d^2 \end{cases}$$

where p , c , d and h are constants. The constants values are set to those in Table IV which make V_{ij} an unbounded repulsive potential field possessing all five necessary properties stated in section VI. Collision Avoidance.

Fig. 8 shows the result of running model (11) with the stationary obstacles shown in Table V. The figure shows how the stationary obstacles are avoided and how the formation then continues its trajectory and restores its shape once the obstacles have been passed. Fig. 9 shows the shortest distance between agents during the simulation. The figure shows how the distance, in the beginning, is stable at four meters. When encountering an obstacle, it drops to about one and a half meters before returning to a stable four meters distance after passing the obstacles.

VIII. DISCUSSION

A. Trajectory Tracking

The goal for the first subproject was achieved. It is apparent from Fig. 3 that the agent can successfully find and follow the

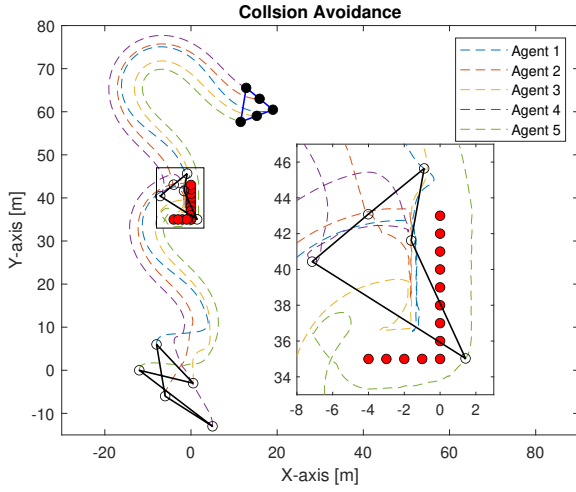


Fig. 8. Five agents avoiding collision with obstacles while following a path in a formation.

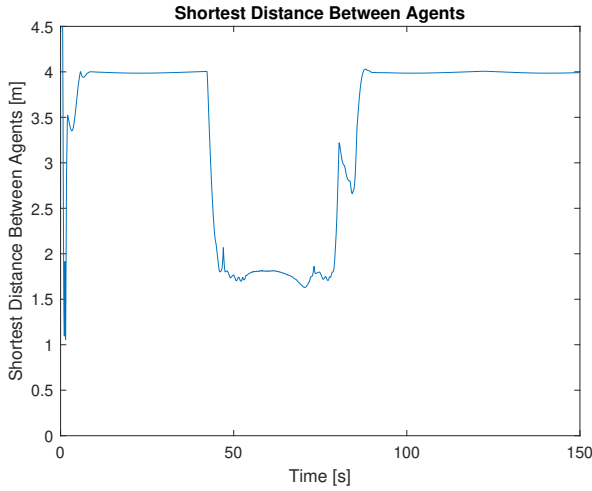


Fig. 9. Plot of the shortest distance between the agents while avoiding collision.

simple and smooth path defined in Table I. Furthermore, the model is also capable of following a non-simple path and a speed that is not constant. The capability to track a varying speed is critical in a practical implementation as there are many reasons to dynamically adjust the speed, for example avoiding obstacles.

Fig. 3 shows how there is a slight discontinuity in the tracking error 5 seconds into the simulation. This is due to the desired angular velocity $u_{2,d}$ not being a smooth curve. At $t = 5$, $u_{2,d}$ switches from $\frac{\pi}{10}$ to $-\frac{\pi}{10}$ which causes the irregularity. The reason that there is no noticeable discontinuity at $t = 15$ is that the system is almost stable. When the state is stable, the error will not diverge from zero regardless of the desired path.

B. Formation Control

The goal for the second subproject was to have 5 agents following a path while holding the formation of an equilateral triangle. Fig. 6 and 7 shows how this was successfully

achieved. The simulation also allows for dynamically rotating the formation which adds more flexibility and applicability of the model. The model used for the simulation also allows for adding more agents by extending the vectors and the graph Laplacian in model (9). It is also possible to change the desired formation by changing the values in Table III and the incidence matrix (13).

The stability of the model could be improved by adding more edges in the graph and updating the incidence matrix (13) correspondingly. This would give the agents more information about how they are required to move in order to achieve the desired formation, which with a double integrator model would imply a shorter settling time.

There are however a few problems with the model which can not be seen in the simulation. The first problem is how there is no actual connection between the moving agents and the desired path. The desired path is extracted from simulating the mean of the agents' starting positions as an agent modelled with the nonholonomic model (1) and has no connection to the actual moving agents. As a result, the model is fragile and will fail to correctly respond to when the agents are not behaving as simulated. If the mean of the model were to be changed outside the expected conditions, there would be a slight discrepancy between the expected trajectory and the actual trajectory, which would never be adjusted. The result would be an error that would continuously increase until the model broke down.

The second problem is how the agents in the model are driven by the double integrator model (7) instead of the desired nonholonomic model (1). The simulation is run in such way, that the path should be possible to follow for the nonholonomic model. This is based on the assumption that the double integrator model can be transformed into the nonholonomic model. This transformation is not actually done, and thus there is no definite proof that the model would work for the intended nonholonomic model. If the simulated model is to be applied to a real AGV both of these problems needs to be solved.

C. Collision Avoidance

The goal for the third subproject was successfully achieved. Fig. 7 shows how the agents never collide with each other, and Fig. 8 shows the agents avoiding collision with the static obstacles. Fig. 8 also shows how the formation is restored and continues along the desired trajectory after passing the obstacles.

As with the second subproject, the model can easily be extended with more agents, different formation or more obstacles. More agents and different formations can be achieved the same way as for the second subproject. Adding more static obstacles can be done by adding more columns to the repulsion matrix (12). It is also possible to model the obstacles as non-static obstacles, i.e. obstacles which are influenced by the agents. Modelling the obstacles as non-static would require the obstacles to be driven by the model by adding them as agents not part of the formation but part of the incidence matrix (13). Using non-static obstacles can lead to a more

dynamic simulation, but may be a dangerous assumption, as it is not always true that obstacles will avoid the agents. It is of more interest to consider moving static obstacles, i.e. moving obstacles which are not influenced by the agents, since that is more applicable to reality. Moving static obstacles could be introduced by simulating an independent system which controls the position of the static obstacles.

By changing the values of the constants in the potential function (10), it is possible to change the effect of the repulsion field. Increasing d and p increases the area of influence and the repulsion force within the area. The constants h and c ensure that all the requirements proposed by [10] for the field are fulfilled with the desired d and p . It is also possible to use a different potential field as long as it satisfies all the requirements from [10]. One useful potential field design would be a field which quickly rises to a high potential near the edges of its area of influence. Such a field would be less flexible than the field used in this project, but might be preferable in some instances. For example, if the agents had a large frame extending far from the sensor sensing the objects.

D. Further Work

There are many aspects of this project which can be improved with further work. Mainly, the transformation from the double integrator model to the nonholonomic model needs to be mathematically proved and simulated. More research is required to determine if it is a feasible option to design the control systems using double integrator models and then convert them back to the nonholonomic model. Furthermore, the nonholonomic model itself could be improved by making it into a more realistic model. For example, by considering the number and positions of the wheels, the friction between the wheels and the surface and eventual slopes.

A critical flaw with the model (9) is how there is no connection between the moving agents and the desired path. If the results of this project were to be applied to a practical scenario such as using physical AGVs this would undeniably result in a growing error. Further work needs to ensure that there is a correlation between the formation and the desired path. This could be done by simulating the driving of the formation at the same time as the formation control, instead of precomputing.

It would be a good idea to improve how the obstacles are modelled. In this project, all obstacles were modelled as points with a potential field. As can be seen in Fig. 8 this is a sufficiently powerful representation of generic obstacles if enough points are added together. The problem is that this requires modelling a lot of point obstacles in order to build a detailed testing environment. Modelling many point obstacles will require a lot of computing power, which will be expensive and slow to simulate. Furthermore, there is no direct conversion of generic obstacles to point obstacles, which is a problem when applying the model to a physical AGV. A better solution would be to model the obstacles as generic polygons and calculating virtual point obstacles at the closest point to each agent. Simulating it this way would require less computing power and also be closer to how a physical AGV

would sense obstacles which increases the applicability of the model.

Finally, the models could be extended from the realm of AGVs into the realm of Autonomous Aerial Vehicles (AAVs). The nonholonomic differential-drive model (1) can be extended into an aerial vehicle model, such as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ \sin(\phi) \\ 0 \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u_2 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_3, \quad (15)$$

where ϕ is the vertical angle and u_3 is the vertical angle velocity. The double integrator models (9) and (12) can easily be extended into aerial space. Model (9) can be extended by extending the size of \bar{v} and \bar{p} to \mathbb{R}^{3N} and two more rows and columns to the matrix. The repulsion matrix (12) can be extended by calculating β_{ij} in the repulsion field (10) using the x, y, z coordinates. What needs to be studied is how trajectory tracking can be realised in a three-dimensional space and if it is possible to transform the double integrator model into the aerial vehicle model (15).

IX. CONCLUSION

In this project, the focus has been on studying cooperative control of autonomous ground vehicles. The project was divided into three subprojects, trajectory tracking, formation control and collision avoidance. The main goal was to have five agents hold a formation and follow an arbitrary path, while avoiding collision with each other and static obstacles.

The kinematic model of the autonomous ground vehicles was chosen as a nonholonomic model. The nonholonomic model is hard to control since it is nonlinear, and therefore a transformation into chained form was made. Transforming to chained form made it easier to control and design the control system. Trajectory tracking was accomplished by converging a tracking error, \bar{x}_e , to zero. When the error converged to zero, the system converged to the desired state and trajectory tracking was achieved.

A displacement based model was used to accomplish formation control of the system. A displacement based model is based on aligning the agents to a desired formation in a local coordinate system and then binding that system to a global coordinate system. This was done by designing the system using a double integrator model, which can then be transformed into the desired nonholonomic model.

The goal for collision avoidance was to avoid collision among the agents and with static obstacles placed along a predetermined path. After avoiding collision, the formation should reform and continue along the desired path. This was achieved by placing a repulsive potential field around all obstacles and agents. By defining the repulsive field as an unbounded potential field no collision was guaranteed.

Finally, simulations were made for all subprojects. The simulations showed how all the goals for the subprojects were achieved, which also implied that the main goal was achieved.

ACKNOWLEDGMENT

The authors would like to thank the supervisor, Fei Chen, for all the support given along the process of this project.

REFERENCES

- [1] S. Boric, E. Schiebel, C. Schlogl, M. Hildebrandt, C. Hofer, D. M. Macht *et al.*, “Research in autonomous driving—a historic bibliometric view of the research development in autonomous driving,” *International Journal of Innovation and Economic Development*, vol. 7, no. 5, pp. 27–44, 2021.
- [2] P. Ögren, D. Anisi, D. Berglund, D. Dimarogonas, H. Gustavsson, L. Hedlin, J. Hedström, X. Hu, H. Johansson, K. F. Katsilieris, V. Kaznov, P. Lif, M. Lindhé, U. Nilsson, M. Persson, M. Seeman, P. Svenmarck, and J. Thunberg, “Results from the project aures: Autonomous ugv-system for reconnaissance and surveillance,” p. 7, 2009.
- [3] S. Walimbe, “The role of autonomous unmanned ground vehicle technologies in defense applications,” *Aerospace Defense Technology Magazine*, 2020.
- [4] S. Nord. (2022, Apr) ”Automated vehicles are more than self-driving cars”. [Online]. Available: <https://www.ri.se/en/what-we-do/our-areas/automated-vehicles>
- [5] M. Akif and S. Geivald, “Cooperative control of autonomous ground vehicles,” Bsc. Thesis, KTH, School of Electrical Engineering and Computer Science (EECS), Stockholm, Sweden, 2021.
- [6] B. Jie Lu and M. Bettar, “Trajectory tracking, formation control and obstacle avoidance for autonomous ground vehicles,” Bsc. thesis, KTH, School of Electrical Engineering and Computer Science (EECS), Stockholm, Sweden, 2020.
- [7] Z. Qu, *Cooperative Control of Dynamical Systems: Applications to Autonomous Vehicles*. London, UK: Springer London, 2009.
- [8] R. M. Murray and S. S. Sastry, “Steering nonholonomic systems in chained form,” 1991.
- [9] K.-K. Oh, M.-C. Park, and H.-S. Ahn, “A survey of multi-agent formation control,” *Automatica*, vol. 53, pp. 424–440, 2015.
- [10] 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, 2008.
- [11] D. V. Dimarogonas and K. H. Johansson, “Further results on the stability of distance-based multi-robot formations,” *2009 American Control Conference*, pp. 2972–2977, 2009.
- [12] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*, ser. Princeton series in applied mathematics. Princeton, New Jersey: Princeton University Press, 2010.

Controlling Autonomous Baker Robot Using Signal Temporal Logic and Control Barrier Functions

Marcus Allen and Gustav Bernpaintner

Abstract—Autonomous systems are slowly moving into the mainstream with things like self driving cars and autonomous robots in storage facilities already in use today. The aim of this project is to simulate a virtual bakery with a baker-robot (agent) that is able to complete recipes within strict deadlines.

Signal temporal logic (STL) is used to define instructions that can be understood by the agent. In order to carry out these instructions, a control barrier function (CBF) is used. CBFs are time and state dependent, are used to describe the desired behavior of the agent, and are designer made. If the CBF corresponding to the task is non-negative from beginning to end during the task, the task has been completed successfully.

A virtual robot was used in this project and was tasked with moving to and staying in different areas, which represents picking up and dropping off ingredients, all whilst staying within the boundaries of the bakery. The focus of this work is on completing the large amount (10+) of sequential tasks required to complete a recipe. The CBF remained positive during the task, and the task was completed successfully.

Sammanfattning—Autonoma system börjar ta mer och mer plats i vardagen med saker som självkörande bilar och autonoma robotar i lagerlokaler som redan används idag. Syftet med det här projektet är att simulera ett virtuellt bageri med en bagarrobot (agent) som kan laga recept under strikta tidskrav.

Signal temporal logic (STL) används för att definiera instruktioner som kan förstås av agenten. För att genomföra dessa instruktioner korrekt används en control barrier function (CBF). CBF:er är tids- och tillståndsberoende, används för att beskriva agentens önskade beteende, och är skapade av en designer. Om CBF:en är positiv från början till slut under uppgiftens gång så har uppgiften genomförts som önskat.

En virtuell robot användes i det här projektet och fick i uppdrag att flytta till och stanna inom olika områden, vilket representerar att plocka upp och lämna ingredienser, allt medan den vistas inom bageriets gränser. Fokus för detta arbete ligger på att slutföra den stora mängd (10+) av sekventiella uppgifter som krävs för att laga ett recept. CBF:en var positiv under hela uppgiften, och uppgiften genomfördes framgångsrikt.

Index Terms—autonomous systems, signal temporal logic, control barrier function, quadratic programming

Supervisors: Maria Charitidou

TRITA number: TRITA-EECS-EX-2022:126

I. INTRODUCTION

The world is constantly moving toward more automation. The use of autonomous household robots, such as autonomous vacuum cleaners and lawn mowers, is increasing in homes due their convenience, reliability and accessibility. The use of autonomous agents in storage facilities enable faster and cheaper services. The recent introduction of self driving cars on the streets has raised the stakes. Every aspect of these au-

tonomous machines needs to work perfectly to avoid disastrous consequences.

One such aspect of automation is to have robots (agents) complete given tasks while avoiding unwanted behaviors. Specifically, this work aims to create a control strategy for an autonomous robot baker (the agent) operating in a virtual bakery that allows it to complete a basic cooking recipe (the task). The task is the conjunction of several subtasks, such as moving ingredients and staying within the bounds of the bakery. The recipe to be completed is a Swedish kladdkaka.

There are multiple ways of both defining the task to be completed and making sure the task is executed correctly. In this work, signal temporal logic (STL) is used to specify the tasks to be completed. STL is chosen because of its temporal aspect since timing is essential when completing recipes. To ensure that the tasks are completed in a satisfactory manner, Control Barrier Functions (CBFs) are used, both because they work well with STL, and also because there is a well-established foundation of work on the subject. To enable the use of CBFs, the STL fragment defined in [1] is used.

Earlier work on the subject of STL and CBFs has focused on aspects such as the generation of control input that is valid in continuous time but generated in discreet time [2], displaying "a good trade-off between computational efficiency and expressivity" for STL/CBF-based control strategies [3], and collaboration between multiple agents [1]. This paper focuses specifically on the execution of a set of tasks in a consecutive manner.

Many tasks take the form "do task A for some amount of time, then once task A is done do task B for some amount of time, etc". For example, this describes the task of completing a cooking recipe, where the order in which the steps are performed often is equally important as ensuring that all the subtasks are completed – the oven needs to be turned on before anything else is done, and mixing before putting ingredients in the mixer would not make any sense. The way to express such tasks used in this work is to encode the desired start time of every consecutive subtask after the end time of the previous subtask, with enough time between the two instants to allow the agent to satisfy the task within the given constraints.

II. CONCEPTUAL OVERVIEW

The process used to ensure that the agent completes its task is described in this section. Section II will begin with a short overview of the process, after which each step is explained more in-depth. Finally, a summary is given which ties all the subprocesses together. This section does not contain any mathematical definitions but rather attempts to intuitively explain

the different concepts of the paper. For the mathematics, see Section III.

A. Overview

When an agent's task has been determined, the first step to completing it is to define it precisely in logical terms. In the context of this report, this means that once a recipe has been chosen, the recipe and its subtasks (such as mixing ingredients or delivering the finished baked goods) need to be defined using STL. The next step is to define the task's CBF, which requires all of the task's subtasks to be defined as candidate CBFs. Once all the subtasks are defined, the main task is defined as the conjunction of its subtasks. Finally, the control input for the agent is calculated.

B. Defining the Task – STL

The first thing to do after deciding on a task is to define it clearly and unambiguously. In this paper, the language for doing this is *signal temporal logic*. STL is a *logical* language used to describe the region within which the agent should be, and the time interval during which the agent should be within the aforementioned region. For a formal definition of STL, see [4]. In this paper, only the STL fragment defined in [1] is used. Specifically, disjunctions are not used because they result in convex predicate functions (see Section II-D), which is not allowed, see [1]. This STL fragment can be used to model tasks such as staying within a region during a specific time interval and moving to an area at some point during a specific time interval, which is all the agent requires in this work. One limitation is that it cannot be used to avoid areas, which is why any desired obstacle avoidance has to be implemented using other methods.

C. Mathematically Defined Subtasks – Candidate CBFs

When working with automatic processes, there are two important aspects according to [5]: *liveness*, which is a guarantee that the desired result will eventually happen, and *safety*, which is the guarantee that undesirable results will not happen.

A CBF is a function that mathematically describes a task and is a function of both state and time, as it describes both *what* needs to be done, as well as *when* and *how* it needs to be done. As long as the CBF is positive from the start time to the end time, both liveness and safety are guaranteed, meaning that the task is completed as desired. Thus, the control objective is to keep the CBF corresponding to the task non-negative for the entire duration of the task.

The CBF for the main task is defined as the conjunction of all its subtasks' CBFs. The CBFs of these subtasks are referred to as *candidate CBFs* in this paper, as in [3], but note that they are mathematically defined in the same way as the main task's CBF. A candidate CBF can either be constructed as a conjunction of other candidate CBFs (if the subtask itself has subtasks of its own), or its creation can be divided into two parts: a *predicate function* that is a-priori known from the STL-definition of the subtask and which describes *what* the task is, and a *gamma function*, which describes *when* and *how* the task should be completed and is designer made.

Once all the candidate CBFs have been designed, the value of the main task's CBF is a continuously differentiable under-approximation of the minimum operator applied on all of its candidate CBFs at any given time and state.

D. What to do – Predicate Functions

A *predicate* is a statement that is either true or false. An example predicate is "The agent is inside a circle with radius r and center c ." Either the agent is inside the circle or it isn't. Every STL task has at least one predicate (or a conjunction of multiple predicates) which has to be true for the task to be satisfied. However, it isn't enough for the candidate CBFs to know if a task is being satisfied or not; it is also necessary to know how well (or poorly) it is being satisfied. This grade of satisfaction is called *space robustness* [6], or just robustness.

The value of the predicate (true or false) is obtained by the evaluation of a function, called predicate function, as follows: the predicate is true if the value of the function is non-negative and false otherwise. Given the robustness metric introduced in [6], the robustness of a given predicate is also computed with respect to the predicate function of the predicate at a given state. In the context of this work, the state is the position of the agent and the robustness is equal to the value of the predicate function.

Take for example the earlier predicate, "The agent is inside a circle with radius r and center c ," illustrated in Fig. 1. If the agent, represented by a star, is at the center of the circle (1 in the figure) the robustness is equal to the radius of the circle, whereas if the agent is just barely inside the circle (2) the robustness is still positive but closer to zero. In the same way, if the agent is barely outside of the circle (3) the robustness is negative but close to zero, and when it is one circle radius away from the circle (4) the robustness is negative the radius of the circle.

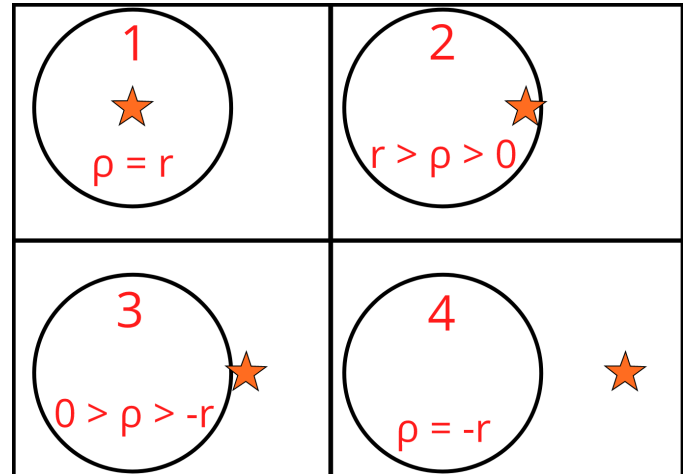


Fig. 1. Four different robustness states illustrated. The agent is represented by a star, and ρ is the robustness (although the predicate function has the same value). As the agent travels further away from the center of the target circle, the robustness decreases.

E. When to do it – Gamma Functions

In Section II-D we discussed how the spatial requirements of a given subtask were fused in the definition of its candidate

CBF. Nevertheless, in order to ensure the satisfaction of each subtask within the given deadlines, a time-varying term needs also to be considered in the definition of the candidate CBF. As a result, the candidate CBF for a given subtask is defined as the sum of two terms, a state-dependent function, i.e., the predicate function, and a time-varying function which ensures the satisfaction of the task, if the candidate CBF remains positive within the interval of satisfaction of the given subtask. This time-varying function is called a *gamma function*.

The effect of the gamma function on the candidate CBF can be illustrated with an example, see Fig. 2. The example uses the same predicate as in Section II-D. However, now an additional time aspect is added: "The agent must be inside a circle with radius r and center c from time t_{start} , represented by frame 3 in the figure, to some ending time t_{end} ." The black circle has radius r and center c . The gray circle with varying radius but the same center c is the circle specified by the candidate CBF, i.e. the circle with respect to both the predicate function *and* the gamma function.

In general, the candidate CBF's area will have the same shape as that in the predicate. This has to do with how the predicate function and candidate CBF are defined, see Section III. As long as the agent, represented with a star, stays within the black circle the predicate function is positive. For the candidate CBF to be positive, however, the agent instead has to stay within the gray circle.

Notice how, at time zero (1 in Fig. 2), the gray candidate CBF circle is bigger than the black predicate circle. This is because the predicate does not yet have to be true. Then, after some time has passed (2), the gray candidate CBF circle has shrunk, forcing the agent closer to the black circle. This is at a time instant closer to t_{start} than in (1). Finally, the gray circle becomes slightly smaller than the black predicate circle (3) at the time when the predicate needs to be satisfied. The reason the gray circle becomes smaller than the black is that the gamma function can be (and has been in the example) designed such that it forces the predicate to be true with a certain robustness, something that is relevant for real-world applications where exact measurements and movements are impossible.

Essentially, by adding a gamma function to the candidate CBF, the agent gets a larger area in which it can move when the predicate does not yet need to be satisfied. As time passes, this area contracts, forcing the agent closer and closer to the predicate circle, until the time when the predicate needs to be satisfied, at which point the gray candidate CBF circle will be fully within the black predicate circle. The candidate CBF is positive as long as the agent stays within the gray circle. Remember that liveness is guaranteed as long as the candidate CBF stays positive. Thus, the candidate CBF, in this case, represents the area in which the agent is free to move at some time instant.

F. Calculating control input - Optimization

Once the candidate CBFs for all the subtasks have been constructed, the CBF that describes the original task is the conjunction of all the candidate CBFs. This CBF is then used

in a quadratic programming problem that is solved to get the next control input for the agent, see Section III-E.

G. Summary

In summary, the first step is to define the task. This is accomplished by defining its subtasks in STL, after which the task is the conjunction of its subtasks. The next step is to express the task mathematically, which is done by defining candidate CBFs for its subtasks. Every CBF is a function that describes *what* needs to be done, *how well* it is being done, and *when it needs to be done*. These three things are done using predicates, the idea of robustness, and gamma functions respectively. The CBF of the original task is the conjunction of all the candidate CBFs. Once this is done, an optimization problem is created using the task's CBF, evaluated at some state and time, which is solved for the next control input for the agent. Fig. 3 shows a visual representation of how the different processes are connected.

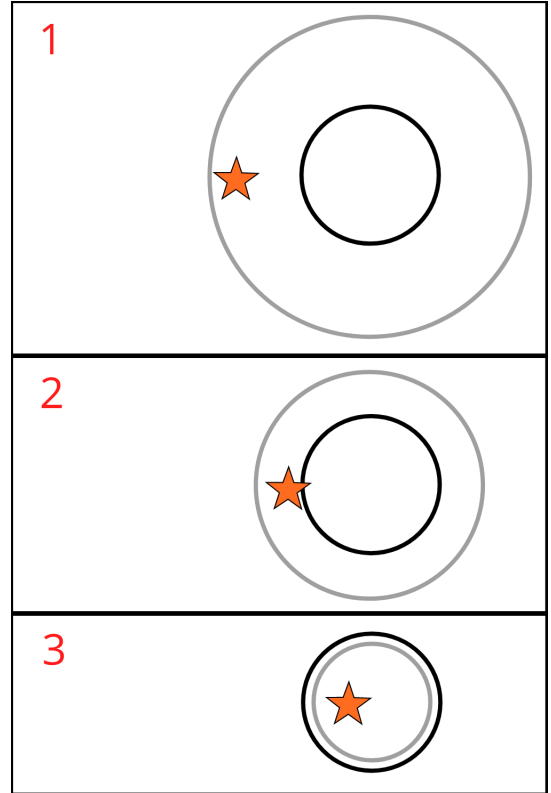


Fig. 2. The agent is represented by a star. As long as it stays within the gray circle, the candidate CBF remains positive and liveness is guaranteed. As long as the agent is inside the black circle, the predicate function is positive. As time passes, the gray candidate CBF circle constricts, forcing the agent to move toward the black predicate circle.

III. THEORY

A. STL Formulations and Syntax

A signal temporal logic formula, ϕ , characterizes the desired behavior of a dynamical system. In this paper we consider the

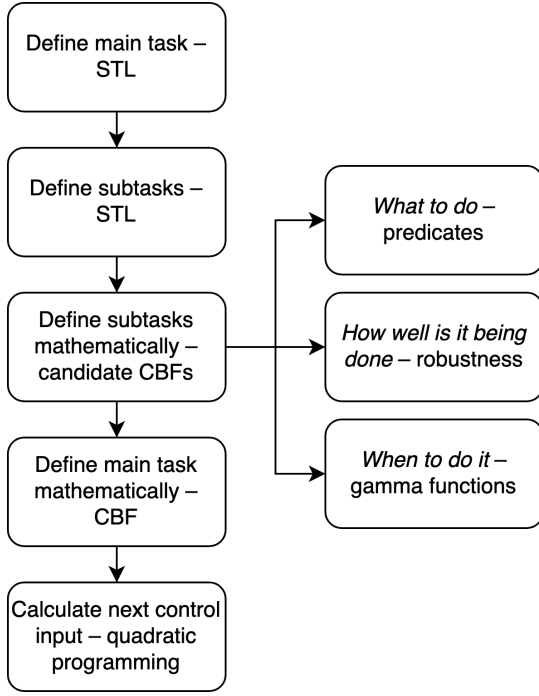


Fig. 3. A flowchart showing how the different processes and tools are connected.

STL fragment defined by [1],

$$\varphi := \top \mid \mu \mid \neg\mu \mid \varphi_1 \wedge \varphi_2 \quad (1a)$$

$$\phi := \mathcal{G}_{[a,b]}\varphi \mid \mathcal{F}_{[a,b]}\varphi \mid \varphi_1 \mathcal{U}_{[a,b]}\varphi_2 \mid \phi_1 \wedge \phi_2, \quad (1b)$$

where φ_1 and φ_2 are STL formulas of the form (1a), ϕ_1 and ϕ_2 are STL formulas of the form (1b). The expressions $\mathcal{G}_{[a,b]}$, $\mathcal{F}_{[a,b]}$ and $\mathcal{U}_{[a,b]}$ denote the "global", "future" and "until" operators, respectively, over the time interval $[a, b]$, where $0 \leq a \leq b < \infty$. The variable μ is the predicate which is determined by the value of a predicate function $h(\mathbf{x})$ as

$$\mu = \begin{cases} \top \text{ (True)}, & h(\mathbf{x}) \geq 0 \\ \perp \text{ (False)}, & h(\mathbf{x}) < 0. \end{cases} \quad (2)$$

The STL semantics used in this project are further detailed with the following definition from [3], where $(\mathbf{x}, t) \models \phi$ denotes ϕ being satisfied by a signal $\mathbf{x} : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$ at time t .

Definition 1: (STL Semantics)

$$\begin{aligned} (\mathbf{x}, t) \models \mu & \Leftrightarrow h(\mathbf{x}(t)) \geq 0 \\ (\mathbf{x}, t) \models \neg\phi & \Leftrightarrow \neg((\mathbf{x}, t) \models \phi) \\ (\mathbf{x}, t) \models \phi_1 \wedge \phi_2 & \Leftrightarrow (\mathbf{x}, t) \models \phi_1 \wedge (\mathbf{x}, t) \models \phi_2 \\ (\mathbf{x}, t) \models \mathcal{G}_{[a,b]}\phi & \Leftrightarrow \forall t_1 \in [t+a, t+b], (\mathbf{x}, t_1) \models \phi \\ (\mathbf{x}, t) \models \mathcal{F}_{[a,b]}\phi & \Leftrightarrow \exists t_1 \in [t+a, t+b] \text{ s.t. } (\mathbf{x}, t_1) \models \phi \\ (\mathbf{x}, t) \models \phi_1 \mathcal{U}_{[a,b]}\phi_2 & \Leftrightarrow \exists t_1 \in [t+a, t+b] \text{ s.t. } (\mathbf{x}, t_1) \models \phi_2 \\ & \quad \wedge \forall t_2 \in [t, t_1], (\mathbf{x}, t_2) \models \phi_1. \end{aligned}$$

The temporal operator $\mathcal{G}_{[a,b]}\phi$ (always) means that ϕ should be satisfied at all times over the time interval $[a, b]$. The operator $\mathcal{F}_{[a,b]}\phi$ (eventually) means that ϕ should be satisfied at some time instant in the time interval $[a, b]$. The operator

$\phi_1 \mathcal{U}_{[a,b]}\phi_2$ (until) means that ϕ_1 should be satisfied at all times from a until t_1 when ϕ_2 is satisfied, $a \leq t_1 \leq b$.

B. Robustness

Robustness is used as a measurement for the satisfaction of a subtask. The *robustness degree* r_i corresponding to the formula ϕ_i denotes the measured value of ϕ_i 's robustness. The variable r_i is a real number that is positive if ϕ_i is satisfied and negative if ϕ_i is violated. The higher the value of r_i is, the better ϕ_i is satisfied, and vice versa. The robustness degree is defined in [6] as "a real number associated with a property-behavior pair, based on, roughly speaking, the distance between the behavior and the (boundary of) the set of all behaviors that satisfy the property. This measure is more positive when the behavior is deeper inside the set of satisfying behaviors and more negative the further is the behavior outside that set." In the context of this work, the property is the STL formula to be satisfied, and the behavior is the state signal, $\mathbf{x}(t)$. Given a predicate μ , a signal $\mathbf{x} \in \mathbb{R}^n$, and STL formulas ϕ , ϕ_1 and ϕ_2 , the robustness semantics are defined by [6] as

$$r^\mu(\mathbf{x}, t) = h(\mathbf{x}(t)) \quad (3)$$

$$r^{\neg\phi}(\mathbf{x}, t) = -r^\phi(\mathbf{x}, t) \quad (4)$$

$$r^{\phi_1 \wedge \phi_2}(\mathbf{x}, t) = \min(r^{\phi_1}(\mathbf{x}, t), r^{\phi_2}(\mathbf{x}, t)) \quad (5)$$

$$r^{\mathcal{G}_{[a,b]}\phi}(\mathbf{x}, t) = \min_{t_1 \in [t+a, t+b]} r^\phi(\mathbf{x}, t_1) \quad (6)$$

$$r^{\mathcal{F}_{[a,b]}\phi}(\mathbf{x}, t) = \max_{t_1 \in [t+a, t+b]} r^\phi(\mathbf{x}, t_1) \quad (7)$$

$$r^{\phi_1 \mathcal{U}_{[a,b]}\phi_2}(\mathbf{x}, t) = \max_{t_1 \in [t+a, t+b]} \min(r^{\phi_2}(\mathbf{x}, t_1), \min_{t_2 \in [t, t_1]} r^{\phi_1}(\mathbf{x}, t_2)). \quad (8)$$

C. Control Barrier Functions

A control barrier function is used in the control design to ensure that the tasks of a problem are fulfilled within given time constraints. The control barrier function $\mathbf{b}(\mathbf{x}, t)$ needs to be positive for every $t \geq 0$ for all subtasks to be satisfied. The CBF of a problem consists of a conjunction of candidate CBFs $\mathbf{b}_i(\mathbf{x}, t)$, where each $\mathbf{b}_i(\mathbf{x}, t)$ corresponds to a subtask within the problem. As in [3] the control barrier function is defined as

$$\mathbf{b}(\mathbf{x}, t) = -\ln \left(\sum_{i=1}^p \exp(-\mathbf{b}_i(\mathbf{x}, t)) \right), \quad (9)$$

which is a smooth under-approximation of the minimum operator $(\min_{i \in \{1, \dots, p\}} \mathbf{b}_i(\mathbf{x}, t))$ that has to be used since the min-operator usually is not a differentiable function. Specifically, for a conjunction of p candidate CBFs $\mathbf{b}_i(\mathbf{x}, t)$, [3] also shows that

$$-\ln \left(\sum_{i=1}^p \exp(-\mathbf{b}_i(\mathbf{x}, t)) \right) \leq \min_{i \in \{1, \dots, p\}} \mathbf{b}_i(\mathbf{x}, t), \quad (10)$$

which guarantees that if $\mathbf{b}(\mathbf{x}, t) \geq 0$ then $\mathbf{b}_i(\mathbf{x}, t) \geq 0$ for all $i \in \{1, \dots, p\}$. The CBF is further developed by the authors in [1] through the implementation of a "deactivation policy" used to reduce its restrictiveness when a high amount of tasks

are given. Each subtask ϕ_i has a corresponding deactivation-function, $o_i(t)$, which is an integer-valued function defined by [7] as

$$o_i(t) = \begin{cases} 1, & t \in T_i \\ 0, & t \notin T_i, \end{cases} \quad (11)$$

for any $t \geq 0$, where T_i is the time interval where $\mathbf{b}_i(\mathbf{x}, t)$ contributes to $\mathbf{b}(\mathbf{x}, t)$, deactivating $\mathbf{b}_i(\mathbf{x}, t)$ when its subtask has been satisfied. Authors in [7] defined T_i as

$$T_i = \begin{cases} [0, b_i], & \text{if } \phi_i = \mathcal{F}_{[a_i, b_i]} \varphi_i \\ (0, b_i), & \text{if } \phi_i = \mathcal{G}_{[0, b_i]} \varphi_i \\ [0, a_i) \cup (a_i, b_i], & \text{if } \phi_i = \mathcal{G}_{[a_i, b_i]} \varphi_i. \end{cases} \quad (12)$$

In this paper, we consider a modified definition where $T_i = (0, b_i)$ for both cases where $\phi_i = \mathcal{G}_{[0, b_i]} \varphi_i$ and $\phi_i = \mathcal{G}_{[a_i, b_i]} \varphi_i$, since it allowed for easier implementation of the deactivation policy. The deactivation policy is utilized in the control barrier function $\mathbf{b}(\mathbf{x}, t)$ as following

$$\mathbf{b}(\mathbf{x}, t) = -\ln \left(\sum_{i=1}^p o_i(t) \exp(-\mathbf{b}_i(\mathbf{x}, t)) \right). \quad (13)$$

1) *Candidate Control Barrier Functions:* Consider a function $\mathbf{b}_i(\mathbf{x}, t)$ defined over the time interval $[t_0, t_1]$, and a set $\mathcal{C}(t)$ defined as

$$\mathcal{C}(t) = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{b}_i(\mathbf{x}, t) \geq 0\} \quad (14a)$$

$$\partial\mathcal{C}(t) = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{b}_i(\mathbf{x}, t) = 0\} \quad (14b)$$

$$\text{Int}(\mathcal{C}(t)) = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{b}_i(\mathbf{x}, t) > 0\}, \quad (14c)$$

where $\partial\mathcal{C}(t)$ is the boundary of the set and $\text{Int}(\mathcal{C}(t))$ is the interior of the set. The function $\mathbf{b}_i(\mathbf{x}, t)$ is a candidate CBF if the following holds, defined by [3].

Definition 2: (Candidate Control Barrier Function) A differentiable function $\mathbf{b}_i : \mathcal{D} \times [t_0, t_1] \rightarrow \mathbb{R}$, where $\mathcal{D} \subseteq \mathbb{R}^n$, is a candidate control barrier function if for each $\mathbf{x}_0 \in \mathcal{C}(t_0)$, there exists an absolutely continuous function $\mathbf{x} : [t_0, t_1] \rightarrow \mathbb{R}^n$ with $\mathbf{x}(t_0) := \mathbf{x}_0$ such that $\mathbf{x}(t) \in \mathcal{C}(t)$ for all $t \in [t_0, t_1]$.

Each candidate CBF $\mathbf{b}_i(\mathbf{x}, t)$ that corresponds to an STL formula ϕ_i with a predicate μ_i is defined in [8] as

$$\mathbf{b}_i(\mathbf{x}, t) = -\gamma_i(t) + h_i(\mathbf{x}), \quad (15)$$

where h_i is the predicate function connected to ϕ_i and $\gamma_i(t)$, the gamma function, describes the desired temporal behavior for the system. The function $\gamma_i(t)$ is defined by the user and is designed to make sure that the formula ϕ_i is satisfied with at least a desired robustness r_i at time t_i^* , where r_i is a tuning parameter as discussed shortly below.

2) *Temporal Behavior:* For this project we have chosen to express γ_i as a piecewise linear function defined in [8] as

$$\gamma_i(t) = \begin{cases} \frac{\gamma_{i,\infty} - \gamma_{i,0}}{t_i^*} t + \gamma_{i,0}, & t < t_i^* \\ \gamma_{i,\infty}, & t \geq t_i^*, \end{cases} \quad (16)$$

where $\gamma_{i,0}$ and $\gamma_{i,\infty}$ are designer-specified constants that are determined by the desired r_i and t_i^* , for which we have

$$\gamma_{i,0} \in (-\infty, h_i(\mathbf{x}(0))) \quad (17a)$$

$$\gamma_{i,\infty} \in (\max(r_i, \gamma_{i,0}), h_i^{max}) \quad (17b)$$

$$t_i^* = \begin{cases} \tilde{b}_i, & \text{if } \phi_i = \mathcal{F}_{[\tilde{a}_i, \tilde{b}_i]} \varphi_i \\ \tilde{a}_i, & \text{if } \phi_i = \mathcal{G}_{[\tilde{a}_i, \tilde{b}_i]} \varphi_i \end{cases} \quad (17c)$$

$$r_i \in \begin{cases} (0, h_i(\mathbf{x}(0))), & \text{if } t_i^* = 0 \\ (0, h_i^{max}), & \text{if } t_i^* > 0. \end{cases} \quad (17d)$$

We can recognize that (17a) is used to ensure $\mathbf{b}_i(\mathbf{x}(0), 0) > 0$. Further, (17b) and (17d) imply that if $\mathbf{b}_i(\mathbf{x}, t) \geq 0$ then $h_i(\mathbf{x}) \geq r_i$ for $t \geq t_i^*$.

D. Valid Control Barrier Functions

Considering a control barrier function $\mathbf{b}(\mathbf{x}, t)$ defined over the time interval $[t_0, t_1]$ we have the following definition by the authors in [3].

Definition 3: (Valid Control Barrier Functions) A CBF $\mathbf{b}(\mathbf{x}, t)$ is defined as a valid control barrier function if there exists a locally Lipschitz continuous class \mathcal{K} function α such that, for all $(\mathbf{x}, t) \in \mathcal{C} \times [t_0, t_1]$,

$$\sup_{\mathbf{u} \in \mathcal{U}} \frac{\partial \mathbf{b}(\mathbf{x}, t)^T}{\partial \mathbf{x}} (f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}) + \frac{\partial \mathbf{b}(\mathbf{x}, t)}{\partial t} \geq -\alpha(\mathbf{b}(\mathbf{x}, t)), \quad (18)$$

where the supremum of the left-hand side of (18) with respect to \mathbf{u} should be at least equal to $-\alpha(\mathbf{b}(\mathbf{x}, t))$. Being a locally Lipschitz continuous class \mathcal{K} function means that α has the attributes of being continuous, monotonically increasing and $\alpha(0) = 0$. From this we have the following lemma proven by [9].

Lemma 1: Let α be a locally Lipschitz continuous class \mathcal{K} function and $v : [t_0, t_1] \rightarrow \mathbb{R}$ be an absolutely continuous function. If $\dot{v}(t) \geq -\alpha(v(t))$ for every $t \in [t_0, t_1]$, and $v(0) \geq 0$, then $v(t) \geq 0$ for all $t \in [t_0, t_1]$.

This lemma is utilized for the control barrier function in (18) to ensure that $\mathbf{b}(\mathbf{x}, t) \geq 0$ by letting $\dot{\mathbf{b}}(\mathbf{x}, t) \geq -\alpha(\mathbf{b}(\mathbf{x}, t))$.

E. Generating Control Input

A control input \mathbf{u} can be generated by solving the optimization problem

$$\min_{\mathbf{u}} \mathbf{u}^T \mathbf{u}, \quad (19)$$

with (18) as a constraint. The control input $\mathbf{u} \in \mathcal{U}$, where the set $\mathcal{U} = \{\mathbf{u} \in \mathbb{R}^n | \mathbf{u} \models \text{Eqn. (18)}\}$ is a non-empty set of inputs and $\mathbf{b}(\mathbf{x}, t)$ is a valid CBF. The goal of (19) is to minimize $\|\mathbf{u}\|$.

IV. METHOD

A. System State and Dynamics

In this work, we consider a two-dimensional system defined as

$$\dot{\mathbf{x}} = \begin{bmatrix} p_x & p_y \end{bmatrix}^T, \quad (20)$$

where p_x and p_y denote the agent's x - and y -coordinates respectively, giving the agent's position \mathbf{x} . The dynamics of the system are defined as

$$\dot{\mathbf{x}} = \mathbf{u} \quad (21a)$$

$$\mathbf{u} = [u_x \quad u_y]^T, \quad (21b)$$

where $\dot{\mathbf{x}}$ is the velocity of the agent and u_x, u_y are the x - and y -velocities given by the control input \mathbf{u} , respectively. The boundaries $|u_x| \leq 15$ and $|u_y| \leq 15$ were set on the velocity so that the agent moves at reasonable speeds inside the bakery. The system state was updated in the simulations according to the equation

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{u}\Delta t, \quad (22)$$

where the time step was chosen as $\Delta t = 0.2$ seconds.

B. Task Specifications

The agent's baking process was broken down into five steps: gathering the required ingredients, mixing the ingredients together, putting the mix in the oven, dropping off the baked cake at a delivery point, and moving to an end zone. The act of picking something up or putting something down, for example an ingredient, was defined as a " $\mathcal{G}_{[a,b]}$ "-task where the agent is told to stay within the designated area for the ingredient for five seconds, where five seconds represent the time it takes to pick up or put down the ingredient. Note that all times in this work are chosen arbitrarily and not necessarily representative of the time requirements of a real kladdkaka recipe.

The problem was designed so that the agent is only allowed to carry one thing at a time. The bakery was defined as a two-dimensional rectangular area centered in the origin with width $d_w = 40$ and height $d_h = 40$. All of the bakery's designated areas for ingredients, mixer, oven, etc, are circular areas defined by their center coordinates and radius, seen in table I.

TABLE I
DEFINITION OF AREAS IN THE BAKERY

Area	Center coordinates (x, y)	Radius	Area number
Butter	(-17.5, 17.5)	2.5	1
Sugar	(-12.5, 17.5)	2.5	2
Eggs	(-7.5, 17.5)	2.5	3
Flour	(-2.5, 17.5)	2.5	4
Baking Powder	(2.5, 17.5)	2.5	5
Chocolate	(7.5, 17.5)	2.5	6
Water/Milk	(-17, 3)	3	7
Mixer	(-12, -16)	4	8
Oven	(16, 16)	4	9
Delivery Point	(17, -17)	3	10
End Zone	(0, 17)	3	11

The main task ϕ was defined as

$$\phi := \phi_{bound} \wedge \phi_{bake}, \quad (23)$$

where ϕ_{bound} is a constantly active subtask that tells the agent to always stay within the boundaries of the bakery during the whole simulation and ϕ_{bake} is the subtask that tells the agent to complete the recipe, which will be further detailed shortly. To

formulate the subtask ϕ_{bound} in STL, the following predicates, predicate functions and formulas were defined

$$h_{i,bound}(\mathbf{x}) \models \mu_{i,bound} \Leftrightarrow h_{i,bound}(\mathbf{x}) \geq 0 \quad (24a)$$

$$h_{1,bound}(\mathbf{x}) = (p_{c,x} + \frac{d_w}{2}) - p_x \quad (24b)$$

$$h_{2,bound}(\mathbf{x}) = p_x - (p_{c,x} - \frac{d_w}{2}) \quad (24c)$$

$$h_{3,bound}(\mathbf{x}) = (p_{c,y} + \frac{d_h}{2}) - p_y \quad (24d)$$

$$h_{4,bound}(\mathbf{x}) = p_y - (p_{c,y} - \frac{d_h}{2}) \quad (24e)$$

$$\varphi_{1,bound} := \mu_{1,bound} \quad (24f)$$

$$\varphi_{2,bound} := \mu_{2,bound} \quad (24g)$$

$$\varphi_{3,bound} := \mu_{3,bound} \quad (24h)$$

$$\varphi_{4,bound} := \mu_{4,bound}, \quad (24i)$$

where each $\varphi_{i,bound}$ -formula corresponds to staying within a wall of the bakery, d_w and d_h are the width and height of the bakery, respectively, p_x and p_y are the x - and y -coordinates of the agent, respectively, $p_{c,x}$ and $p_{c,y}$ are the x - and y -coordinates of the bakery's center respectively. The bakery's dimensions and center point were set as $d_w = d_h = 40$, $p_{c,x} = p_{c,y} = 0$, the agent's initial state is also set in the origin, $\mathbf{x} = \vec{0}$ at time $t = 0$. From (24f), (24g), (24h) and (24i) the STL formula ϕ_{bound} was defined as

$$\varphi_{\Omega} := \varphi_{1,bound} \wedge \varphi_{2,bound} \wedge \varphi_{3,bound} \wedge \varphi_{4,bound} \quad (25a)$$

$$\phi_{bound} := \mathcal{G}_{[t_{start}, t_{end}]} \varphi_{\Omega}, \quad (25b)$$

where $t_{start} = 0$ is the starting time of the simulation and t_{end} is the end time of the simulation.

The agent was tasked with baking a kladdkaka, which requires five different ingredients from the bakery that have to be put in the mixer in the following order: butter, eggs, sugar, chocolate and flour. Since the agent only is allowed to carry one thing at a time, it has to go to the mixer in between getting new ingredients. Gathering the ingredients is formulated to the agent as: go to Area 1 (Butter) and stay there for five seconds, then go to area 8 (Mixer) and stay there for five seconds, then go to area 3 (Eggs) and stay there for five seconds, etc. After mixing all the ingredients together the agent was tasked with putting the cake in the oven (by staying within area 9 for five seconds), then dropping the cake off at the delivery point (by staying within area 10 for five seconds) and lastly going the end zone (Area 11). The subtask of going to a designated area $\varphi_{i,o}$ was defined in STL as

$$h_{i,o}(\mathbf{x}) \models \mu_{i,o} \Leftrightarrow h_{i,o}(\mathbf{x}) \geq 0 \quad (26a)$$

$$h_{i,o}(\mathbf{x}) = r_i - \|\mathbf{x} - \mathbf{c}_i\| \quad (26b)$$

$$\varphi_{i,o} := \mu_{i,o} \quad (26c)$$

where i is the area number, r_i and \mathbf{c}_i are the radius and center of area i , respectively. The task of picking up or putting something down is defined as $\mathcal{G}_{[a,b]} \varphi_{i,o}$, where the time interval $[a, b]$ is five seconds long. The subtask ϕ_{bake} was defined with the following STL formulas, where t_{end} was set

to 125 s

$$\phi_1 := \mathcal{G}_{[5,10]} \varphi_{1,o} \quad (27a)$$

$$\phi_2 := \mathcal{G}_{[15,20]} \varphi_{8,o} \quad (27b)$$

$$\phi_3 := \mathcal{G}_{[25,30]} \varphi_{3,o} \quad (27c)$$

$$\phi_4 := \mathcal{G}_{[35,40]} \varphi_{8,o} \quad (27d)$$

$$\phi_5 := \mathcal{G}_{[45,50]} \varphi_{2,o} \quad (27e)$$

$$\phi_6 := \mathcal{G}_{[55,60]} \varphi_{8,o} \quad (27f)$$

$$\phi_7 := \mathcal{G}_{[65,70]} \varphi_{6,o} \quad (27g)$$

$$\phi_8 := \mathcal{G}_{[75,80]} \varphi_{8,o} \quad (27h)$$

$$\phi_9 := \mathcal{G}_{[85,90]} \varphi_{4,o} \quad (27i)$$

$$\phi_{10} := \mathcal{G}_{[95,100]} \varphi_{8,o} \quad (27j)$$

$$\phi_{11} := \mathcal{G}_{[105,110]} \varphi_{9,o} \quad (27k)$$

$$\phi_{12} := \mathcal{G}_{[115,120]} \varphi_{10,o} \quad (27l)$$

$$\phi_{13} := \mathcal{F}_{[120,t_{end}]} \varphi_{11,o} \quad (27m)$$

$$\phi_{bake} := \bigwedge_{j=1}^{13} \phi_j. \quad (27n)$$

C. Temporal Constraints and Tuning

The candidate control barrier function $\mathbf{b}_i(\mathbf{x}, t)$ for each of the subtasks containing predicate functions had their values of $\gamma_{i,0}$ and $\gamma_{i,\infty}$ tuned depending on the time constraints of the subtask. The candidate CBFs used to satisfy ϕ_{bound} were defined as

$$\mathbf{b}_1(\mathbf{x}, t) = -\gamma_1(t) + h_{1,bound}(\mathbf{x}) \quad (28a)$$

$$\mathbf{b}_2(\mathbf{x}, t) = -\gamma_2(t) + h_{2,bound}(\mathbf{x}) \quad (28b)$$

$$\mathbf{b}_3(\mathbf{x}, t) = -\gamma_3(t) + h_{3,bound}(\mathbf{x}) \quad (28c)$$

$$\mathbf{b}_4(\mathbf{x}, t) = -\gamma_4(t) + h_{4,bound}(\mathbf{x}). \quad (28d)$$

The candidate CBFs used for satisfying ϕ_{bake} were defined as

$$\mathbf{b}_5(\mathbf{x}, t) = -\gamma_5(t) + h_{1,o}(\mathbf{x}) \quad (29a)$$

$$\mathbf{b}_6(\mathbf{x}, t) = -\gamma_6(t) + h_{8,o}(\mathbf{x}) \quad (29b)$$

$$\mathbf{b}_7(\mathbf{x}, t) = -\gamma_7(t) + h_{3,o}(\mathbf{x}) \quad (29c)$$

$$\mathbf{b}_8(\mathbf{x}, t) = -\gamma_8(t) + h_{8,o}(\mathbf{x}) \quad (29d)$$

$$\mathbf{b}_9(\mathbf{x}, t) = -\gamma_9(t) + h_{2,o}(\mathbf{x}) \quad (29e)$$

$$\mathbf{b}_{10}(\mathbf{x}, t) = -\gamma_{10}(t) + h_{8,o}(\mathbf{x}) \quad (29f)$$

$$\mathbf{b}_{11}(\mathbf{x}, t) = -\gamma_{11}(t) + h_{6,o}(\mathbf{x}) \quad (29g)$$

$$\mathbf{b}_{12}(\mathbf{x}, t) = -\gamma_{12}(t) + h_{8,o}(\mathbf{x}) \quad (29h)$$

$$\mathbf{b}_{13}(\mathbf{x}, t) = -\gamma_{13}(t) + h_{4,o}(\mathbf{x}) \quad (29i)$$

$$\mathbf{b}_{14}(\mathbf{x}, t) = -\gamma_{14}(t) + h_{8,o}(\mathbf{x}) \quad (29j)$$

$$\mathbf{b}_{15}(\mathbf{x}, t) = -\gamma_{15}(t) + h_{9,o}(\mathbf{x}) \quad (29k)$$

$$\mathbf{b}_{16}(\mathbf{x}, t) = -\gamma_{16}(t) + h_{10,o}(\mathbf{x}) \quad (29l)$$

$$\mathbf{b}_{17}(\mathbf{x}, t) = -\gamma_{17}(t) + h_{11,o}(\mathbf{x}). \quad (29m)$$

The tuning values $\gamma_{i,0}$ and $\gamma_{i,\infty}$ for all candidate CBFs are given in table II.

TABLE II
TUNING VALUES CHOSEN FOR THE SIMULATION

i	$\gamma_{i,0}$	$\gamma_{i,\infty}$
1	0	0.1
2	0	0.1
3	0	0.1
4	0	0.1
5	-35	0.1
6	-105	0.1
7	-175	0.1
8	-245	0.1
9	-315	0.1
10	-385	0.1
11	-455	0.1
12	-525	0.1
13	-595	0.1
14	-665	0.1
15	-755	0.1
16	-805	0.1
17	-875	0.1

D. Control Input

The control input \mathbf{u} was generated by solving the following optimization problem:

$$\min_{\mathbf{u}} \mathbf{u}^T \mathbf{u} \quad (30a)$$

$$\frac{\partial \mathbf{b}(\mathbf{x}, t)^T}{\partial \mathbf{x}} \mathbf{u} + \frac{\partial \mathbf{b}(\mathbf{x}, t)}{\partial t} \geq -\alpha(\mathbf{b}(\mathbf{x}, t)). \quad (30b)$$

The function $\alpha(\mathbf{b}(\mathbf{x}, t))$ was chosen as a linear function $\alpha(\mathbf{b}(\mathbf{x}, t)) = \alpha \cdot \mathbf{b}(\mathbf{x}, t)$, where α was tuned and set as $\alpha = 1$. The derivatives in (30b) are given by

$$\frac{\partial \mathbf{b}(\mathbf{x}, t)}{\partial \mathbf{x}} = \frac{\sum_{i=1}^p o_i(t) \exp(-\mathbf{b}_i(\mathbf{x}, t)) \frac{\partial \mathbf{b}_i(\mathbf{x}, t)}{\partial \mathbf{x}}}{\sum_{i=1}^p o_i(t) \exp(-\mathbf{b}_i(\mathbf{x}, t))} \quad (31a)$$

$$\frac{\partial \mathbf{b}(\mathbf{x}, t)}{\partial t} = \frac{\sum_{i=1}^p o_i(t) \exp(-\mathbf{b}_i(\mathbf{x}, t)) \frac{\partial \mathbf{b}_i(\mathbf{x}, t)}{\partial t}}{\sum_{i=1}^p o_i(t) \exp(-\mathbf{b}_i(\mathbf{x}, t))}. \quad (31b)$$

E. Simulations

The simulations followed algorithm 1.

Algorithm 1: Simulation algorithm

```

Define agent's initial state;
Define areas;
Define predicate functions;
Define  $\gamma$  functions;
Define control barrier functions;
while time  $t < t_{end} = 125$  seconds do
    Calculate control input;
    Update state;
    Increment time;
end

```

The simulations and plots were made in Python using Numpy, Scipy and Matplotlib. The function `scipy.optimize.minimize` with solver option `method='SLSQP'` was used to solve the optimization problem. All code written for the project can be found at https://github.com/GBernpaintner/robot_baker.

V. RESULTS

The results for the control barrier function $b(x, t)$ and trajectory of the agent gained from the simulations can be seen in fig. 4 and fig. 5, respectively.

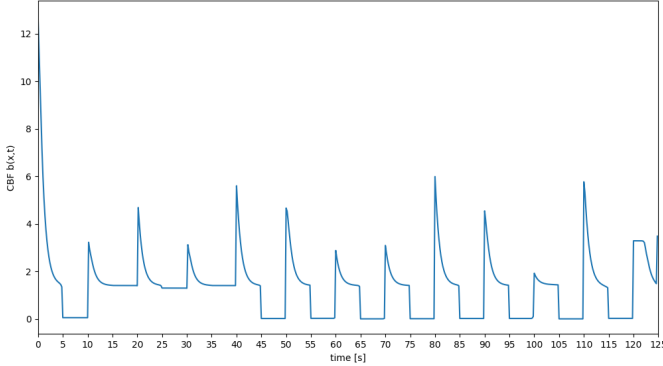


Fig. 4. Value of the control barrier function, $b(x, t)$, during the simulation.

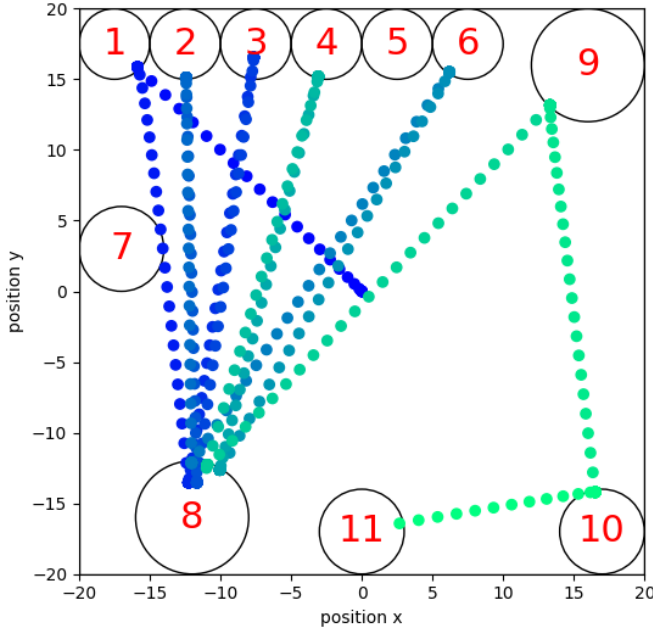


Fig. 5. Full trajectory of the agent represented by the dots. The agent starts at the dark blue dot in the center, and moves from dark blue to light green. Each area is indicated by its area number.

Fig. 6 is included to closely show that the CBF's value is positive at all times. This means that the original task of baking a kladdkaka was satisfied.

VI. DISCUSSION

The goal of the project was to have the agent "complete the recipe for a Swedish kladdkaka", a recipe which requires five different ingredients from the bakery. Since we only allowed the agent to carry one ingredient at a time it needed to go to the mixer in between getting each ingredient, which meant that the agent got two new subtasks for every ingredient added. This led to a lot of subtasks that had to be completed by the agent and tuned by us. The main task gave the agent a total

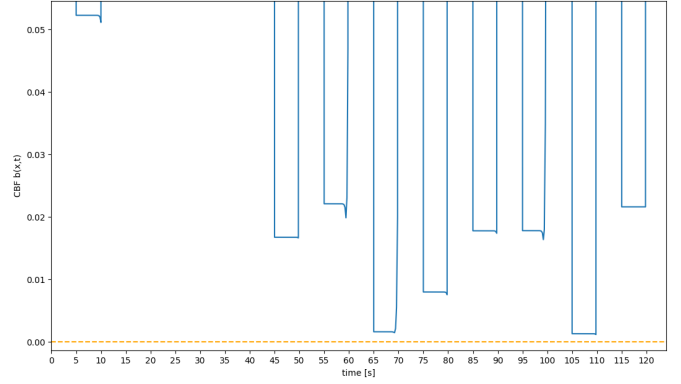


Fig. 6. Zoomed in graph of the control barrier function's value, $b(x, t)$. Orange line marks $b(x, t) = 0$.

of 17 subtasks – four subtasks for staying within the bakery, ten subtasks for gathering and mixing ingredients and three subtasks for putting the mix in the oven, putting the cake in the delivery zone and going to the end zone.

A. Comments on Results

As we can see from fig. 4 and fig. 6 the CBF was positive during all times of the simulation which means that the agent was able to satisfy all subtasks within the given time intervals, thus it was able to complete the kladdkaka-recipe. If we study the figures closer we can see that the CBF has several dips where it stays flat. The time intervals of each dip span over the same time intervals as the ϕ_i -subtasks where the agent is tasked to stay within a designated area for five seconds, for example, if we look at the first dip in fig. 4, we can see that it spans from 5 s to 10 s, which corresponds to the subtask $\phi_1 := \mathcal{G}_{[5,10]} \varphi_{1,\circ}$. The CBF stays flat during these subtasks because the agent is standing still within the designated area during the given time interval. Since the agent is inside the area, it does not have to move anymore to satisfy the predicate, and therefore stays still. This means that the predicate function's value, and in turn the corresponding candidate CBF's value, stays constant.

At the end of each flat dip, we see an instantaneous increase in the CBF. This happens when the subtask corresponding to the dip has been fully satisfied and it's candidate CBF gets deactivated. Since the CBF is a smooth under-approximation of the minimum value of all the active candidate CBFs, it means that if the subtask with the smallest candidate CBF is deactivated, the CBF will take on a new value which is at most equal to the *next* lowest candidate CBF.

Looking at fig. 6 we can also see that all of the dips do not go down to the same value. Since we've set $\gamma_{i,\infty} = 0.1$ for all subtasks it means that they are satisfied with a robustness of at least 0.1, but some tasks are satisfied more than others. If a dip has a higher value it means that the corresponding subtask is better satisfied, which in this project means that the agent is further within the designated area. We can see this if we look at fig. 5, where the agent stops just within the circle boundaries of some circles (i.e areas 2, 4, 6 and 9) and further within other circles (i.e areas 1, 3 and sometimes 8). This implies

that the dip corresponding to subtask $\phi_1 = \mathcal{G}_{[5,10]}\varphi_{1,o}$ should have a higher value than the dip corresponding to subtask $\phi_7 = \mathcal{G}_{[65,70]}\varphi_{6,o}$, which is true if we compare the CBF's values between time intervals $[5, 10]$ and $[65, 70]$ in fig. 6.

B. Future Work

For future work, it would be interesting to add more recipes, and also to add obstacles and more agents to the bakery. Adding more recipes is not difficult as long as the required ingredients are defined in the bakery, but problems could arise if the recipes contain many ingredients. As mentioned earlier, each added ingredient adds two subtasks to the main problem, which can quickly lead to a lot of candidate CBFs and γ -functions that all need to be tuned individually. Even though they are tuned separately, they all have overlapping effects on the CBFs and the agent's trajectory. This means that when we tune a γ -function we also need to take in account the tuned values of the other γ -functions, which gets increasingly difficult the more subtasks we have.

Adding obstacles and more agents to the bakery would require implementations of obstacle avoidance and collision avoidance, which would also affect tuning and possibly make it harder.

VII. CONCLUSIONS

The objective of this project was to create a virtual autonomous bakery where the user could give a recipe, formulated with a set of STL tasks, to an autonomous baker robot and have the robot complete the recipe within a given time frame. The group has provided a virtual environment with a functioning bakery, where a user specifies STL tasks for the baker robot. Most of the STL tasks consist of the agent entering and staying within different regions within the bakery to gather and transport different objects required for completing the recipe. The agent's main task was to complete the recipe for a Swedish "kladdkaka," which the results showed it was able to do, by satisfying all given subtasks within their respective time intervals.

ACKNOWLEDGMENTS

The authors would like to thank their supervisor Maria for all the help during the project. Without her, it wouldn't have been possible.

REFERENCES

- [1] 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.
- [2] G. Yang, C. Belta, and R. Tron, "Continuous-time signal temporal logic planning with control barrier functions," in *2020 American Control Conference (ACC)*, 2020, pp. 4612–4618.
- [3] 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.
- [4] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," in *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, Y. Lakhnech and S. Yovine, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 152–166.

- [5] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*, 2019, pp. 3420–3431.
- [6] A. Donzé and O. Maler, "Robust satisfaction of temporal logic over real-valued signals," in *Formal Modeling and Analysis of Timed Systems*, K. Chatterjee and T. A. Henzinger, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 92–106.
- [7] M. Charitidou and D. V. Dimarogonas, "Barrier function-based model predictive control under signal temporal logic specifications," in *2021 European Control Conference (ECC)*. IEEE, 2021, pp. 734–739.
- [8] L. Lindemann and D. Dimarogonas, "Barrier function based collaborative control of multiple robots under signal temporal logic tasks," *IEEE Transactions on Control of Network Systems*, vol. PP, pp. 1–1, 08 2020.
- [9] H. K. Khalil, *Nonlinear systems; 3rd ed.* Upper Saddle River, NJ: Prentice-Hall, 2002, the book can be consulted by contacting: PH-AID: Wallet, Lionel. [Online]. Available: <https://cds.cern.ch/record/1173048>

CONTEXT C – PART I

LEARNING IN DYNAMICAL SYSTEMS

POPULAR DESCRIPTION

The magical box of finance

The general consensus in the world is that in order to succeed in the financial market you need to put in a lot of time. The 10000 hour rule is quoted in lots of cases. A lot of analyzing and reading financial magazines is required. Keeping up to date every waking hour is necessary. What if the widely accepted fact that a lot of time is needed just isn't true anymore? What if computers can make those 10000 hours into a mere matter of minutes?

Making predictions in the financial market is a very difficult job with tremendously huge risk and just as big of a reward. Trading in financial markets is a fast paced activity. For example in high-frequency trading, customers can buy and sell stocks several times per second. This demands decision making where accuracy and speed is the name of the game.

Investors usually spend their working days looking at vast amounts of data from various different markets to make predictions of the stocks or derivatives they are interested in. Because of the complexity of the financial market, this takes a significant amount of time in order to get a somewhat reliable result. Systems like this that are constantly changing are called dynamical systems and the financial market, despite being one of the most complex, is far from the only one.

Dynamical systems appear everywhere, for example: how the population in a country increases; how much rain falls each summer in a specific area; and of course how stocks on financial markets change. Dynamical systems are everywhere in our world and there are a huge amount of people, such as demographers, geologists and economists whose goal is to predict these dynamical systems. However, the work is gradually being moved from humans to computers, which is both quicker and more accurate compared to humans. Just feed this “magical box” with data and it will spit out reasonably reliable predictions within seconds.

SUMMARY OF PROJECT RESULTS

Increasing computing power is paving the way to solving various complex problems, and learning algorithms is one prominent example of a technology benefiting from this. As learning algorithms improve, lots of different fields are seeing potential for increased adoption. Finance is an example where new applications regarding these types of algorithms are embraced. A common goal in finance is accurate price forecasting, another goal could be the ability to recognize patterns in trades or holdings of an adversary. Optimization techniques and machine learning methods are crucial in working towards goals such as these in a world of increasing information flow and complexity of data. The aim of this context is therefore exploring how these techniques and methods can be leveraged on financial and related data, with the purpose of increased profit and saved time.

The project groups in C1 have used machine learning to estimate parameters of financial models, as a means to predict future behavior of the financial market, and its dependence on external factors.

Group C1a estimated parameters for stochastic models of the financial market. Using artificially generated data, the volatility and drift parameters of a real stock could be estimated by means of indirect inference: a simulation based technique, that estimates the parameters through an auxiliary model. More recent parameter estimation methods, such as the two stage approach, were used in conjunction with neural networks to find parameters that generalize well. These models are desirable

since they work better on unobserved data and therefore may improve forecasting performance. The two methods' forecasting performance were also analyzed and compared.

Group C1b produced parametric models of hourly electricity prices over different time periods in order to, at the end of the project, compare the parametric models. To obtain the model parameters the group used data from some of the external factors that affect the electricity prices the most. The past hourly values of the external factors were assembled in a matrix, that with the help of the regression method lasso, was shrunk to a matrix with few non-zero values, that were the model parameters. While the previous values were updated, new forecastings were therefore obtained. While analyzing the results, the statistical performance and the obtained model parameters were compared for the different models. Conclusions of the different statistical performances showed how well the regression model lasso suits for the different time periods. After analyzing the different parameters for the models, the group came to conclusions about which external factors that affect the electricity prices more in specific time periods.

Project group C4a aim was to predict the future value of different stocks using different technical and fundamental parameters. These parameters were then used in different machine learning models such as the LSTM-model. The goal was to find parameters, that indicate whether the stock will go up or down in value. The result group C4a attained was that there did not exist any significant parameters on how the stocks will move in the future (out of the parameters tested). Thus, the core of the report was to discuss the process of doing the analysis of the difference of the different machine learning models; and to discuss which one is best to use in what circumstances. The results obtained gave information whether technical or fundamental analysis is more important in determining the stocks future value.

In project C4b we tackled the problem of reverse engineering trading portfolios with the help of deep learning methods. The portfolios are represented by trading strategies- algorithms that decide whether to buy or sell based on daily price and volume data. By combining Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) architectures we created models able to handle large time-series and adapt to a wide range of potential underlying algorithms. With scarcity of real data in mind, we used a deep learning technique called transfer learning to pre-train a large model on simulated data. Different combinations of architectures and techniques were evaluated on strategies of varying complexity and data in different amounts. The models that have been developed in this project have been shown to be able to identify technical strategies and transfer learning shows promising results when trained on small amounts of data.

Project group C5 studied the models used by a large number of investors, utilising a common portfolio-allocation model to inversely estimate expected return and risk aversion by said investors, assuming the portfolio is optimal. The structure of the problem was exploited using inverse optimization techniques, in order to investigate the models used by investors for their portfolio allocation. The group compared different regularizer penalties (such as comparing the expected return of different investors) to illuminate the advantages and disadvantages of different models. In addition, the group tested the model on self-generated data, in order to analyse the use-cases of such models.

An interesting follow-up project to **project C4b** is to dive deeper into how transfer learning can be leveraged to create models able to learn complex strategies from small amounts of data. In **project C4a** a follow-up project that can be performed is to study if the machine learning method will perform differently if only stocks from a specific category is to be analyzed. And if the result from this method differs from the result in **project C4a**, is it better or worse?

IMPACT ON SOCIETY AND ENVIRONMENT

Data analysis and learning algorithms will play a fundamental part in tomorrow's society. Nearly everything can be modeled as a dynamical system from which information can be retrieved, past behavior analyzed and future actions predicted. This imposes questions of ethical nature regarding the impacts from the services these digital advancements bring.

A digitalized society is built on data. The user consumes a large amount of data but also leaves data behind for others to collect. This enables optimal, intelligent services such as ad targeting and personal recommendations. With the data amount growing exponentially each day, the performance of the data analysis increases even more.

The ability to accurately estimate or predict information, values or behavior of an agent based on simple data in the forms of search history, attention tracking, financial records or web cookies raises issues about personal privacy. As a consumer of a digital service, you are not completely aware or educated about how your data is collected and used. Regulations such as the Right to Be Forgotten or the Right to an Explanation tries to give private consumers the right and possibility to control their data, although the service's dependency on this data, the technological complexity or the the asymmetry of capacity, time and money for a consumer to engage in judicial matters with large companies might make the consumer unwilling to actually enforce or test their rights.

The enormous amount of data that is aggregated and processed results in a security issue, since its potentially valuable information might be of interest to others than intended. The owner of the data then needs to prevent exposing the consumer's data and thus their information, by protecting it from cyberthreats or failures. Severe accidents have already occurred such as sensitive Swedish health records on Vårdguiden 1177 (Swedish healthcare service provider) being openly accessible or Avanza (Swedish stockbroker firm) unawaringly sharing their customers financial data to third-parties such as Facebook. This has direct impacts on individuals. With crucial welfare services such as health care or insurances also being digitized, individuals are now being required to submit sensitive personal information online.

The comfort and mainstreaming of these digital services could also make the user or the society dependent on data hoarding and analysis. It could therefore be difficult to adjust system flaws or security risks if it would result in inferior performance of the services. However, data analysis provides more than comfort. By making optimal recommendations or conclusions for improvement, almost any service could become more efficient but also unlock new services.

Another important use of data is not only to make conclusions from the past, but to make conclusions responsive to new data. The data becomes a learning tool to model systems and construct algorithms taking autonomous decisions: Machine Learning. These decision algorithms can make quick and effective decisions out of complex problems that would otherwise require the attention of a person or not be possible to solve at all. Though these machine learning algorithms can recognize patterns difficult to see for a person, they also work in ways that are largely unknown to those using them.

An algorithm in which the decision process is unknown might have difficulties arguing the reasoning behind a ruling. This means that false positives, and false negatives might be very difficult to identify. Is it possible to avoid incorporating bias in algorithms, making sure it does not make unjust decisions? Additionally, if an algorithm is in control, the question of who bears the responsibility if the decisions are found to be faulty arises. Is it the developer of said algorithms, or the one using them?

One interesting example highlights a class of algorithms used in the criminal justice system in the United States, risk assessment algorithms. These algorithms are designed to predict the chances of the defendant committing future misconducts, using parameters such as the individual's crime history and age, along with other factors. The risk that they will be convicted for a future crime, and the risk that they will fail to appear in court are calculated and translated into a score used to inform the sentencing judge. Proponents arguing the use of these algorithms claim that they bring consistency, accuracy and transparency to criminal sentencing, something that has been found to vary greatly between different judges. Along with their increased use, they have also become more controversial. Critics argue that the biggest issues are: lack of individualization, absence of transparency under trade-secrets claims and possibility of bias in the data.

An example of where decision algorithms are seeing use is the financial world, often seen as a place where morals and ethics are put aside for the purpose of increased profit. Even if this is the case, it is hard to argue about the importance of financial markets for society as a whole and individuals. Many people depend on the workings of markets. In a direct way by saving money in markets and indirectly by association with the company they work at or the house they own. The application of optimization techniques and machine learning methods can here work as an aid in decision making regarding creation of portfolios, trading strategies and identification of patterns with the purpose of financial gain, stability or identification of risks.

Individuals can without much effort get improved financial security by investing in a variety of funds optimized with regards to risk and reward, lessening worry of the future. Automatic trading algorithms can provide liquidity to markets and as time passes also act as a cushion for sudden disturbances in the market, lowering the risk of a market collapse and all its

consequences on society. However, introducing these types of algorithms to markets without enormous amounts of testing can lead to greater market disturbances instead of cushioning them and for individuals financial ruin is not an unlikely scenario.

As machine learning algorithms reach higher complexity and data gathering becomes bigger, our energy usage is increasing. However the potential to solve and optimize numerous problems is better than ever. Algorithm-generated finance portfolios could potentially be easy to optimize or nudge toward trading more environmentally friendly papers, and machine learning gives us the ability to improve resource usage and better our waste management. Unfortunately it is difficult to tell if the increased knowledge but higher energy consumption is a beneficial trade off. That is something time will tell.

All in all, the possibility to learn from data and create algorithms making their own decisions has large potential to impact our society for good. But as with many digital technologies they are applicable to both good and bad uses. The implementation becomes a trade-off between much-needed benefits such as resource efficiency, stable financial markets or increased equality, and the negative impacts or ethical challenges such as integrity, accountability or uncertainty. But with use of caution and both producers and users being adequately educated, these issues could be addressed and managed resulting in the benefits outweighing the risks.

Comparison of Indirect Inference and the Two Stage Approach

Victor Hernadi and Leandro Carocca

Abstract—Parametric models are used to understand dynamical systems and predict its future behavior. It is difficult to estimate the model's parametric values since there are usually many parameters and they are highly correlated. The aim of this project is to apply the method of indirect inference and the two stage approach to estimate the drift and volatility parameters of a Geometric Brownian Motion. This was first done by estimating the parameters of a known Geometric Brownian process. Then, the Coca-Cola Company's stock was used for a five-year forecast to study the estimators' predictive power. The two stage approach struggles when the data does not truly follow a Geometric Brownian Motion, but when it does it produces highly efficient and accurate estimates. The method of indirect inference produces better estimates, than the two stage approach, for data that deviates from a Geometric Brownian Motion. Therefore, it is preferable to use indirect inference over two stage approach for stock price forecasting.

Sammanfattning—Parametriska modeller används för att förstå dynamiska system och förutspå dess framtida beteende. Det är utmanande att skatta modellens parametriska värden eftersom det vanligtvis finns många parametrar och de är ofta starkt korrelerade. Målet med detta projekt är att tillämpa metoderna indirect inference och two stage approach för att skatta drivnings- och volatilitetsparametrarna av en geometrisk Brownsk rörelse. Först skattades parametrarna av en känd Geometrisk Brownsk rörelse. Sedan användes The Coca-Cola Companys aktie i syfte att studera estimatorernas förmåga att förutspå en femårig period. Two stage approach fungerar dåligt för data som inte helt följer en geometrisk Brownsk rörelse, men när datan gör det är skattningarna noggranna och effektiva. Indirect inference ger bättre skattningar än two stage approach när datan inte helt följer en geometrisk Brownsk rörelse. Därför är indirect inference att föredra för aktieprognoser.

Index Terms—Geometric Brownian Motion, Drift, Volatility, Indirect Inference, Two Stage Approach, Parameter Estimation, Stock Price Prediction

Supervisor: Braghadeesh Lakshminarayanan

TRITA number: TRITA-EECS-EX-2022:127

I. INTRODUCTION

The goal of science has always been to make sense of the world around us. To make sense of a system, we formulate models which describe the system's characteristics, but perhaps more importantly: allow for future prediction. The strive for such models has lead to major breakthroughs in all areas of science, however, some systems remain a challenge for us to model accurately. While some systems lend themselves to be modeled accurately with relatively few parameters, other systems may require an infeasible amount of parameters to be modeled with any hope of accuracy. To make matters worse, these parameters can be highly correlated and their correlation

may only be partially known, and some parameters may not even be observable. Despite these challenges, parametric models remain an important topic of research due to their benefits, such as interpretability and not being as dependent on high quality and voluminous data, compared to non-parametric models.

There are many ways to guide the construction of parametric models; the common approach usually consists of two parts: assuming a functional form followed by parameter estimation [1, pp 21]. When assuming a functional form, one may use intuition, a priori knowledge, empirical evidence or any other justification. This gives the constructor the freedom to choose the complexity of their model with the ultimate aim to approximate the real functional form of the system. The next part is to estimate the parameters which define the chosen functional form of the model. To do this, one usually resorts to statistical methods of estimation.

The problem of parameter estimation in statistics involves applying some rule, known as the *estimator*, on sample data to calculate the (in some sense) best guess, known as the *estimate*, of a parameter value, called the *estimand*. When choosing an estimator, there are some desirable properties one should look for, namely unbiasedness, minimum variance, consistency and efficiency. One of the very earliest methods of parameter estimation is the method of moments, which is easy to work with due to its simplicity. However, the estimators produced by the method of moments are often biased and are not guaranteed to be efficient. A popular alternative is the method of maximum likelihood, which attains several of the desirable properties in the limiting sense, but the relevant equations may be difficult to derive and solve (often numerically) for some applications [2, pp 47–54]. Viewing the problem as a regression problem, the most common estimator is the least squares estimator, especially linear least squares for its simplicity and readily available computational tooling. Extensions to the least squares estimators includes techniques of regularization such as ridge regression and lasso. These techniques are particularly useful whenever estimands are highly correlated since they perform feature selection and can reduce the variance of the estimator [1, pp 237–245].

More often than not, the constructed model becomes too unwieldy and difficult to manage analytically. With the advancement in computing, simulation based techniques for parameter estimation is now in common use for these problems. One such technique is indirect inference, which replaces the complex model with a simpler model (almost always misspecified) and then relying on large scale simulation of the complex model to correct for the bias [3, pp 61]. A more modern method is the two stage approach which aims to construct an estimator by

relying on generalizable properties of the model. In contrast to indirect inference, the two stage approach offloads the computational effort of simulation to an initial training phase; this makes it possible to produce estimates without additional computational effort once training is completed [4].

This paper applies the method of indirect inference and the two stage approach to estimate the drift and volatility parameters of a Geometric Brownian Motion (GBM). This is a stochastic differential equation used to model stock prices (or any underlying asset) in the Black–Scholes model. While GBM serves as the canonical example for indirect inference [3, pp 122], the two stage approach has not been used, to the best of the authors’ knowledge, to estimate the drift and volatility of GBM. The two stage approach is an interesting alternative to indirect inference, since it might produce computationally cheap estimates which is ever more important in high frequency trading.

II. BACKGROUND

A. Geometric Brownian Motion

In the second *annus mirabilis* paper, theoretical physicist Albert Einstein laid forth groundbreaking work in statistical mechanics by modeling the random motion of pollen particles in a liquid [5]. This type of continuous-time stochastic process is called a Brownian motion which is mathematically described by a Wiener process, W_t . A Wiener process is defined by the following conditions [6, pp 6]:

- 1) $W_0 = 0$;
- 2) W_t has independent increments;
- 3) W_t has continuous paths;
- 4) $W_t - W_s \sim \mathcal{N}(0, t - s)$ for $s \leq t$.

A Wiener process, by itself, cannot model a stock price since the process is centered around zero and may take on negative values, unlike any real stock price. If we assume that the relative change of a stock price, S_t , is governed by a constant drift and a stochastic part, we may model the change with the following stochastic differential equation (SDE):

$$\frac{dS_t}{S_t} = \mu dt + \sigma dW_t, \quad S_0 = s_0 > 0 \quad (1)$$

where μ is the drift and W_t is a Wiener process and s_0 is some initial stock price. The parameter σ is constant and is known as the volatility and it has a scaling effect on the Wiener process. Whenever a process abides (1), it is said to follow a GBM. It can be shown [6, pp 218], that the solution to (1) is given by:

$$S_t = S_0 e^{(\mu - \frac{\sigma^2}{2})t + \sigma W_t}. \quad (2)$$

It is clear from (2) that $S_t > 0$ for every t and is not centered around zero; this makes it possible to model a stock’s price with GBM.

B. Indirect Inference

Let \mathcal{M} , characterized by the parameter vector θ , be a simulable model describing a dynamic system of the form:

$$y_t = \mathcal{M}(y_{t-1}, x_t, u_t, \theta), \quad t = 1, 2, \dots, T \quad (3)$$

where y_t is a sequence of observable endogenous variables, x_t is a sequence of observable exogenous variables and u_t is a sequence of stochastic errors that are not observable. In theory, it is possible to estimate the true value of θ , denoted by θ_0 , given x_t along with initial values y_0 and u_0 using a maximum likelihood approach. However, such an approach may yield intractable likelihood functions. Instead, the method of indirect inference exploits the simulable property of \mathcal{M} . For a given θ we can produce several, say H , artificial sequences, $\tilde{y}_t^h(\theta)$, using the observed set of exogenous variables. Now, the main idea of indirect inference is to match the artificial sequences with an observed sequence, y_t^0 , through an auxiliary model. This auxiliary model, generally misspecified, has its own parameter vector β that should be more easily estimated using either observed data or simulated data. The indirect estimate, $\hat{\theta}$, is the θ that makes the auxiliary model’s estimate of the observed data, $\hat{\beta} = \beta(y_t^0, \theta)$, as close as possible to the auxiliary model’s estimate of the artificial data:

$$\hat{\beta} = \frac{1}{H} \sum_{h=1}^H \beta(\tilde{y}_t^h, \theta). \quad (4)$$

More formally, the indirect estimate is the θ that minimizes the following quadratic form:

$$\hat{\theta} \leftarrow \arg \min_{\theta \in \Theta} (\hat{\beta} - \tilde{\beta})^T W (\hat{\beta} - \tilde{\beta}) \quad (5)$$

where W is some positive definitive matrix and Θ is some parameter space. It is possible to prove that (5) is a consistent estimator, i.e. $\hat{\theta} \rightarrow \theta_0$ under quite weak conditions [7]. Generally, a suitable auxiliary model is one that has at least as many parameters as the original model and those parameters should be easily estimated. Its most important feature is its ability to distinguish each path, i.e. the auxiliary model does not necessarily have to fit the data well, but instead capture the variability of θ .

C. Two Stage Approach

As with indirect inference, we utilize the simulable model \mathcal{M} , described in (3), to produce artificial data. We decide on a parameter space Θ , presumably a range of values which is of interest, from which we produce artificial data. For each $\theta_i \in \Theta$ where $i = 1, 2, \dots, m$, we realize a sequence \tilde{y}_t^i through simulations. Now the task is to construct a function f such that:

$$f \leftarrow \min_f \frac{1}{m} \sum_{i=1}^m \|\theta_i - f(\tilde{y}_t^i)\|^2. \quad (6)$$

The function f is effectively a map from data to parameters, i.e. $\theta = f(\tilde{y}_t(\theta))$. It is important to note, that the function f is constructed solely with artificial data, and we must assume it generalizes well. The optimization problem (6) is in general difficult to solve due to the high dimensionality of the problem. The two stage approach tries to solve this in two stages. First a compressive stage, which aims to reduce the dimension of the problem. Each sequence \tilde{y}_t^i is compressed into α_t^i where $t = 1, 2, \dots, \tau \ll T$. This can be done in any manner, but the method proposed by the original authors is by fitting an

ARX model to \tilde{y}_t^i which would then give the compression as a sequence of weights. This compression stage can be viewed as finding a function g such that $\alpha_t^i = g(\tilde{y}_t^i)$. Similarly to the auxiliary model in indirect inference, this function's most important feature is to reduce the complexity while still capturing the variability of θ_i . Instead of solving (6) directly, the second stage attempts to solve this with α_t^i instead of \tilde{y}_t^i , i.e. we find a function h such that:

$$h \leftarrow \min_h \frac{1}{m} \sum_{i=1}^m \|\theta_i - h(\alpha_t^i)\|^2. \quad (7)$$

Since $\tau \ll T$, problem (7) should be easier to solve than (6). Any class of functions for h , whether neural networks or any other non-linear function (or even linear), can be employed and found by traditional methods. The summary is that g compresses the data, and then h is fitted onto the compressed data, all in order to reduce the dimensionality of the problem. This means that we have trained a model to predict the parameters for any process that follows (3). Under the assumption that the data generalizes well, any unknown θ from a sequence y_t should be estimable by the composition $h \circ g$, i.e. $\hat{\theta} = h(g(y_t))$. Since all computational effort is only done once – when constructing the estimator on artificial data – new estimates are produced with minimal effort [4].

III. METHODOLOGY

In this section we present how we use indirect inference and the two stage approach to estimate $\theta = [\mu \ \sigma]^T$ from a process that follows GBM.

A. Parameter estimation using indirect inference

We first consider a naive discretization of (1):

$$y_t = (1 + \mu) y_{t-1} + \sigma y_{t-1} \varepsilon_t, \quad \varepsilon_t \sim \text{IID}(0, 1). \quad (8)$$

as the simulable model and then use the easily derivable maximum likelihood estimators of (8) as the auxiliary model's parameters. The parameter estimate is $\beta = [\mu_{\text{ML}} \ \sigma_{\text{ML}}]^T$ where the maximum likelihood estimators are:

$$\mu_{\text{ML}} = -1 + \frac{1}{T} \sum_{t=1}^T \frac{y_t}{y_{t-1}}, \quad (9)$$

$$\sigma_{\text{ML}} = \sqrt{\frac{1}{T} \sum_{t=1}^T \left(\frac{y_t}{y_{t-1}} - (1 + \mu) \right)^2} \quad (10)$$

where we use $\mu = \mu_{\text{ML}}$ in (10). Now the indirect estimate is given by (5) where W is the identity.

Using the same simulable model, we also consider a general autoregressive model of order p , $\text{AR}(p)$, as our auxiliary model:

$$y_t = \sum_{i=1}^p \beta_i y_{t-i} + \varepsilon_t. \quad (11)$$

The auxiliary model's parameter estimate is determined by traditional least squares method which gives us the estimate $\beta = [\beta_1 \ \beta_2 \ \dots \ \beta_p]^T$. As before, the indirect estimate is given by (5) where W is the identity.

B. Parameter estimation using two stage approach

We decide a parameter space $\Theta = \{\mu_k\}_{k=1}^N \times \{\sigma_k\}_{k=1}^M$ and then generate artificial data with the simulable model in (8). First we consider using the maximum likelihood estimator as our compressing stage, giving us the compressed artificial data as $\alpha_i = [\mu_{\text{ML}} \ \sigma_{\text{ML}}]^T$ for $i = 1, 2, \dots, NM$. We solve the optimization problem (7) by employing a multi-output regression neural network. The number of neurons in the input layer must match the dimension of α – two neurons for maximum likelihood – and the hidden layer was chosen to be of size 32, while the output layer consists of two neurons regardless of compression. We use a softmax activation function and a mean squared error (MSE) loss function. Once the neural network has been trained, the two stage approach estimate is determined by composing the two stages, i.e. by first compressing the data and then feeding the compression into the neural network.

We also consider using an $\text{AR}(p)$ model, same as (11), as our compression stage, which compresses the data into the parameters $\alpha_i = [\alpha_1 \ \alpha_2 \ \dots \ \alpha_p]^T$. The neural network is constructed and used in the same manner as before, but now with p neurons in the input layer.

C. Parameter estimation evaluation

A GBM sequence is realized, through simulation, with known μ and σ . Evaluation is done by estimating the parameters of this known sequence using indirect inference and the two stage approach. This is done for all the auxiliary models and compressing functions described previously.

D. Stock price forecasting

To study the forecasting performance of these models, we estimate the drift and volatility of The Coca-Cola Company stock (KO). The estimation is carried out over the time period 2000-2010. Once the drift and volatility are estimated (by either method), we use GBM simulations using the estimated parameters to forecast the stock's price 5 years ahead. Evaluation is done by comparing the forecast with real stock price between 2010 and 2015.

IV. RESULTS AND DISCUSSION

A. Parameter estimation

A known GBM sequence with $\mu = 0.1$ and $\sigma = 0.2$ over a time span $T = 253$, to emulate the stock market's 253 business days, was realized. The distribution of the indirect estimate of μ , using 1000 replications and $H = 1$, is shown in Fig. 1 while Fig. 2 shows the indirect estimate of σ under the same conditions.

The maximum likelihood (ML) estimator performs better than the autoregressive models for both μ and σ as shown in Fig. 1 and Fig. 2. In Table I, we see the superiority of the ML estimator more clearly: it achieves the lowest bias and standard deviation for both μ and σ . The autoregressive models performed almost equally, where the $\text{AR}(3)$ model performed slightly better. The higher bias for σ , using the ML estimator, is almost surely due to the estimator (10) being dependent on

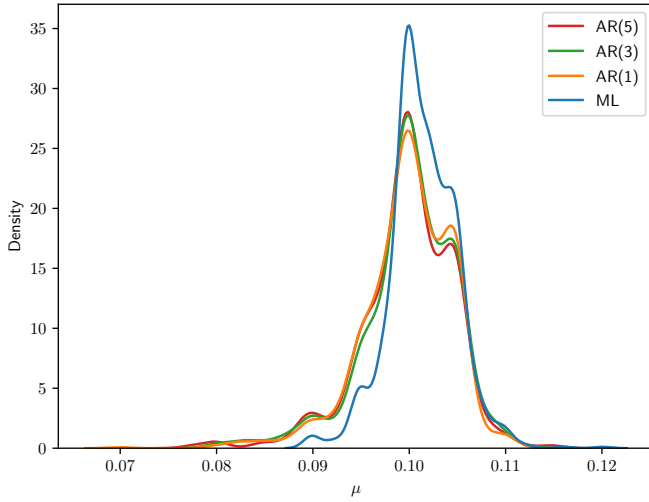


Fig. 1. Distribution of μ estimates over 1000 replications for a maximum likelihood (ML) auxiliary model and $AR(p)$ auxiliary models with $p = \{1, 3, 5\}$.

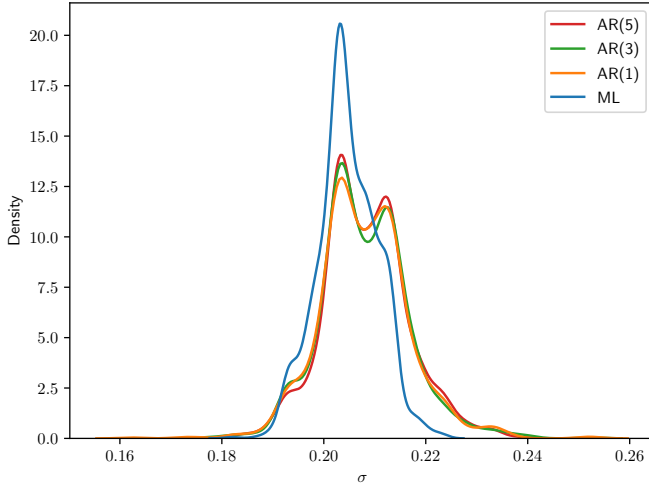


Fig. 2. Distribution of σ estimates over 1000 replications for a maximum likelihood (ML) auxiliary model and $AR(p)$ auxiliary models with $p = \{1, 3, 5\}$.

$\mu = \mu_{ML}$ – the bias from μ_{ML} propagates. Higher bias for σ using AR models is likely due to their inability to model the variability of σ . It is more difficult to estimate the volatility than the drift regardless of auxiliary model, which is supported by the RMSE values; they are almost equal, or equal, to the standard deviation of the drift estimates while they are always higher than the standard deviation of volatility estimates.

The parameter space considered for the two stage approach consists of the drift parameter space $\{-0.5, -0.49, \dots, 0.5\}$ and volatility space $\{0, 0.01, \dots, 0.5\}$, so the entire parameter space is their Cartesian product. For each parameter value, 100 paths are generated with $T = 253$. The two stage approach estimates of the known aforementioned GBM sequence, with $\mu = 0.1$ and $\sigma = 0.2$, are presented in Table II.

Compression with AR models tends to yield poor estimates, compared to indirect inference, as shown in Table II. However, ML compression produces estimates of μ and σ with the smallest bias out of all methods considered. The AR

models consistently perform worse than maximum likelihood estimates across the board, which could indicate their inability to capture the variability of the parameter space for GBM processes.

TABLE I
INDIRECT INFERENCE ESTIMATES.

Aux. model		Mean	Bias	Standard Deviation	RMSE
ML	μ	0.1014	0.0014	0.0034	0.0037
	σ	0.2047	0.0047	0.0058	0.0075
AR(1)	μ	0.0998	-0.0002	0.0047	0.0047
	σ	0.2079	0.0079	0.0088	0.0118
AR(3)	μ	0.0999	-0.0001	0.0048	0.0048
	σ	0.2080	0.0080	0.0085	0.0117
AR(5)	μ	0.0998	-0.0002	0.0048	0.0048
	σ	0.2083	0.0083	0.0081	0.0116

TABLE II
TWO STAGE APPROACH ESTIMATES.

Compression		Estimate	Bias
ML	μ	0.1007	0.0007
	σ	0.2042	0.0042
AR(1)	μ	0.1218	0.0218
	σ	0.2135	0.0135
AR(3)	μ	0.1352	0.0352
	σ	0.2708	0.0708
AR(5)	μ	0.1340	0.0340
	σ	0.2586	0.0586

B. Forecasting

The Coca Cola Company stock (KO) from 2000 to 2015 is shown in Fig. 3. The drift and volatility estimates during the first ten years are shown in Table III.

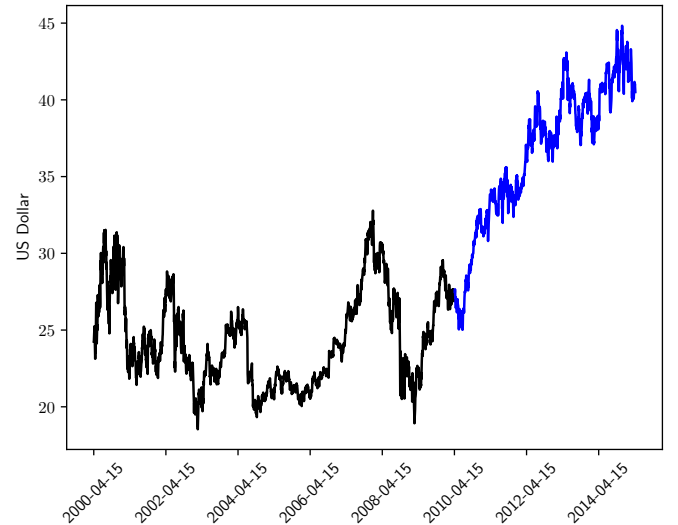


Fig. 3. KO stock from 2000-04-15 to 2015-04-15 where the first 10 years (black) is the training data and the rest (blue) evaluation data.

TABLE III
PARAMETER ESTIMATES OF KO STOCK.

Estimator		Estimate
Indirect Inference ML	μ	0.00016
	σ	0.01469
Two stage approach ML	μ	0.00066
	σ	0.01618
Indirect Inference AR(1)	μ	0.00016
	σ	0.01492
Two stage approach AR(1)	μ	0.05233
	σ	0.19989

Immediately, we can tell the two stage approach with AR(1) produces an unreasonable parameter estimate; such a high drift over 5 years would produce astronomical values and thus we omit the result. The KO stock price in 2010 was \$27.49, and \$40.51 in 2015. Using the estimates, except two stage approach with AR(1), in Table III we project out 1000 possible price trajectories using (8) to gather a distribution of stock prices after five years, shown in Fig. 4. The indirect estimates are least biased and quite similar, while the two stage approach estimate overestimates the price significantly. The two stage approach works under the assumption the artificial data generalizes well; this could be seen when the two stage approach was used to predict parameters of a sequence that actually followed a GBM process. However, the KO stock is *not* a GBM process and the working assumption is violated, which is why we see poor estimates. Meanwhile, indirect inference manages to mitigate this by correcting the estimates with the empirical data.

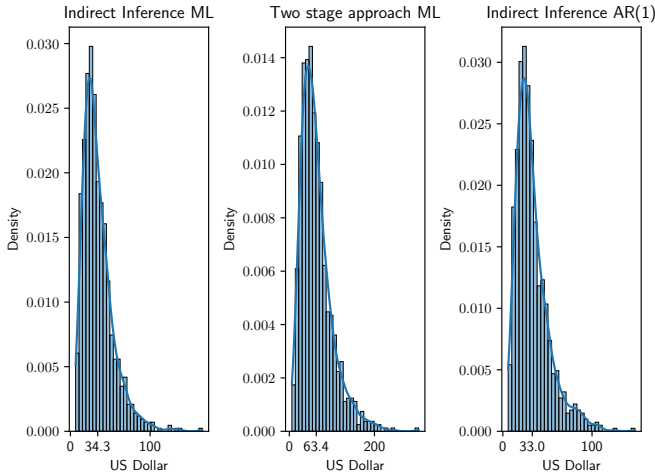


Fig. 4. Distribution of predicted KO stock prices after five years.

V. CONCLUSION AND FUTURE WORK

Although stock prices do not generally follow a GBM and the working assumption of the two stage approach is violated, the two stage approach shows very promising results for estimating parameters of an actual GBM process. Its computational advantage, over indirect inference, at generating these estimates is significant. Therefore, it should still be interesting to further investigate this approach, especially by considering other compression functions and neural network

architectures. Additionally, one could separate each parameter estimate into its own problem, i.e. choosing a compression function and neural network architecture independently for each parameter. This could yield better estimates, especially for σ , since it would be possible to tailor the compression function and architecture to the specific parameter.

For parameter estimation when the data surely follows a GBM process, the two stage approach with an ML compression produces the fastest and the least biased estimates, while both AR and ML auxiliary models for indirect inference produces better estimates for real stock data.

ACKNOWLEDGMENT

The authors would like to thank our supervisor Braghadeesh Lakshminarayanan for his support and guidance.

REFERENCES

- [1] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: with applications in R*, 2nd ed., ser. Springer Text in Statistics. New York, US: Springer, 2021, vol. 103.
- [2] P. K. Sahu, S. R. Pal, and A. K. Das, *Estimation and Inferential Statistics*. New Delhi, India: Springer, 2015.
- [3] C. Gouriéroux and A. Monfort, *Simulation-Based Econometric Methods*, ser. CORE Lectures. New York, US: Oxford University Press, 1996.
- [4] S. Garatti and S. Bittanti, "A new paradigm for parameter estimation in system modeling," *International Journal of Adaptive Control and Signal Processing*, vol. 27, no. 8, pp. 667–687, 2013.
- [5] A. Einstein, "Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen," *Annalen der Physik*, vol. 322, no. 8, p. 549–560, 1905.
- [6] R. F. Bass, *Stochastic Processes*, ser. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge, UK: Cambridge University Press, 2011.
- [7] C. Gouriéroux, A. Monfort, and E. Renault, "Indirect Inference," *Journal of Applied Econometrics*, vol. 8, pp. S85–S118, 1993.

Predictions of Electricity Prices in Different Time Periods Using Lasso

Xue Liu and Harriet Manning

Abstract—When the big data time comes, people also need to keep pace with the times to seek and develop tools that can deal with the vast amount of information. In this project, lasso is applied to build parametric models of electricity prices based on different affecting factors. Thereafter, the models are used to predict the electricity prices 8 days forward for three different time periods. We compare their prediction performances in terms of normalized mean square error (NMSE) and identify dominant factors of the electricity prices in different time periods using lasso. The results show that a model that spans over a 24 hour long period gives the lowest NMSE, followed by one spanning over a two hour long period where the electricity prices are leading up to a peak value. The model that obtains the highest NMSE is from a two hour long period, where the electricity prices have a peak value. Besides, we also analyze potential reasons for the results.

Sammanfattning—När big data-tiden kommer måste även människor hålla jämna steg med tiderna för att söka och utveckla verktyg som kan hantera den stora mängden information. I detta projekt används lasso för att bygga parametriska modeller av elpriser baserade på olika påverkansfaktorer. Därefter används modellerna för att förutsäga elpriserna 8 dagar framåt för tre olika tidsperioder. Vi jämför deras prediktionsprestanda i termer av normaliserat medelkvadratfel (NMSE) och identifierar dominerande faktorer för elpriserna under olika tidsperioder med hjälp av lasso. Resultaten visar att en modell som sträcker sig över en 24 timmar lång period ger lägst NMSE värde, följt av en som sträcker sig över en två timmar lång period där elpriserna leder fram till ett toppvärde. Modellen som får högst NMSE är från en två timmar lång period, där elpriserna har ett toppvärde. Dessutom analyserar vi också potentiella orsaker till resultaten.

Index Terms—Electricity price prediction, linear model, lasso, affecting factors.

Supervisors: Yu Wang

TRITA number: TRITA-EECS-EX-2022:128

I. INTRODUCTION

Electricity is essential for daily life. Electricity prices affect not only private individuals that are e.g., in need of electricity for heating and lighting, but also companies that use it in e.g., productions. For this reason, being able to estimate the prices of electricity can help to reduce the cost for users by regulating their energy consumption at different hours of the day. Different sellers and buyers of energy such as electricity management companies can maximize their profits with the help of reliable predictions of electricity prices.

Since a vast amount of related data is generated daily in the electricity market, it is very time consuming to analyze the electricity market and make predictions. Machine learning

methods can be applied to reduce the cost of analysis and prediction.

The project aims to explore the linear relationship between the Swedish electricity prices and some external factors, which are different types of energy and total production, consumption and net exchange of energy. We assume that electricity prices are linearly dependent on these external factors. The electricity prices depend on many energy sources. As shown in [1], there is a relationship between electricity prices and fossil fuels, e.g., gasoline and heating oil etc. In the article [2], the authors find that there is a relationship between electricity prices and hydro power. Since electricity is mainly produced by renewable sources according to [3], wind power and nuclear power also have an interesting correlation to the electricity prices.

In order to explore more accurate ways to do predictions, different parametric models are identified to predict electrical energy prices 8 days forward in three distinct time periods i.e., the whole day (24 hours), the time period between 8 and 10 o'clock and the time period between 16 and 18 o'clock. The identification of the three parametric models is inspired by [4]. In [4], the authors focus on high frequency intra-day trading periods, which means data in every half hour, and observe that there are different affecting external factors in different intra-day trading periods. Therefore, we choose to detect two shorter time periods, out of the daily 24 hour long period. The detection is based on the change of the most influencing external factors, and thereafter we use available data to identify three parametric models.

The method that is applied in this project is the linear regularization method lasso. It is chosen considering the assumption of a linear relationship between the external factors and the electricity price. Furthermore, lasso does shrinkage and variable selection at the same time which provides predictions that are both accurate and interpretable [5]. This is a quite desirable trade off since the parametric model aims to be used by both professional and non-professional traders e.g., home owners that want to regulate their energy consumption, and will for this reason require interpretability. The predictions of the three identified models are compared in terms of normalized mean square error (NMSE) [6]. However, the calculated NMSE values show that the generated models do not provide an adequate accuracy of prediction. The reasons are analyzed as well in this report.

II. METHOD

A. Lasso

Lasso is a parametric model which assumes a linear relationship between its predictors and responses, which, respectively, are the model's input data and output data. Accordingly to [7], the general form for a linear parametric function is as follows:

$$y = \beta_0 + X\beta_1 + \epsilon, \quad (1)$$

where y is the response vector, with n as the sample size [8]. β_0 and β_1 are the parameters to be identified. ϵ is a noise with mean zero [7]. Additionally, the amount of model parameters are equivalent to the amount of columns in the predictor matrix X , denoted as p . The predictor matrix is $n \times p$ -dimensional [8]. Since the predictor matrix X is $n \times p$ -dimensional and β_1 is p -dimensional, when performing the matrix multiplication $X\beta_1$, each individual column in X is multiplied with one model parameter in β_1 each. The goal of parametric modeling methods is to produce a simplified model where the output depends on parameters which correlate with the output. For this reason, the values of the vector β_1 , the model parameters, will have different magnitudes, due to the different correlations between the corresponding input (external factors) and the output (electricity price). In order to obtain the parameter values, a statistical machine learning method is used. Lasso is a regularization method that can obtain both interpretative and accurate predictions. Mathematically the method is described in equation (2)

$$\begin{aligned} \hat{\beta}_1 = \arg \min & \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_{1j})^2 \\ \text{s.t.} & \sum_{j=1}^p |\beta_{1j}| \leq t \end{aligned} \quad (2)$$

where $\hat{\beta}_1$ stands for the estimation value of β_1 . $i = 1, \dots, n$, $j = 1, \dots, p$. We make the following assumptions to apply lasso. First, the regressors (inputs) are independent. Second, we normalize the regressors. Third, the predictor matrix X is with full rank. In order to enable interpretable results, β_1 is shrunk by the tuning parameter t . Further, t is restrained by the condition $t < t_0$, where $t_0 = \sum |\hat{\beta}_j^0|$, which is the sum of the full least squares estimates, in absolute value, accordingly to [5]. The least squares estimates can be solved with equation (2) without the penalization $\sum_{j=1}^p |\beta_{1j}| \leq t$, obtaining the optimization problem [8]. Since t_0 is a sum of absolute values, its function forms a rotated square, with all its corners on the coordinate axis's, which is the constraint region as can be seen in Fig. 1. The sum of the square errors for equation (1) can be described mathematically by the quadratic function in equation (3).

$$(\beta - \hat{\beta}^0)^T X^T X (\beta - \hat{\beta}^0). \quad (3)$$

The function forms elliptical circles centered around the ordinary least squares (OLS) estimate $\hat{\beta}$. Since the estimator $\hat{\beta}_1$ in the lasso model is shrunk by the tuning parameter, the solutions are located where the elliptical circles meet the constraint region. The sparse solution of the lasso estimator

$\hat{\beta}_1$ can be visually explained by observing that many of the elliptical circles will cut in the corners of the constrained region in Fig. 1 [5]. β_0 is thereafter equivalent to \bar{y} that is the mean value of y according to [9].

According to [9], equation (2) can be written in its equivalent Lagrangian form shown in equation (4),

$$\hat{\beta}_1 = \arg \min \frac{1}{2} \left(\sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_{1j})^2 + \lambda \sum_{j=1}^p |\beta_{1j}| \right) \quad (4)$$

where the regularization paramter λ is depending on the tuning parameter t and both can be estimated using cross validation, accordingly to [5].

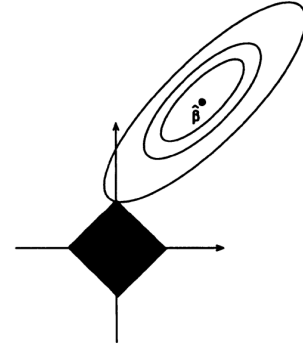


Fig. 1. Sparse solutions of the lasso estimator [5].

B. The training data and test data

Two important concepts that need to be clarified in order to understand how estimations in linear models are conducted are *training data* and *test data*. The linear model described in Section II-A is created with training data, values for the predictors and responses. By inserting the training data in the lasso function, values of $\hat{\beta}_1$ and $\hat{\beta}_0$ are estimated, obtaining the linear model. In order to make predictions, new values of the predictors, called the test data, are inserted to the linear model. There should be a time shift between the parameters in the training data that is equal to the time shift between the test data and the predicted values of y . For example, an estimation with predictions one day forwards has an equivalent time shift of one day between the predictors and responses in the training data as well.

C. NMSE

NMSE (normalized mean square error [6]) is applied in the project to evaluate the statistical performance of the parametric models. This method is applied rather than MSE because the MSE's obtained for the predictions of the different parametric models are not possible to compare, since the MSE's are in different scales due different lengths of data sets. Therefore, it is convincing if a conclusion is drawn based on NMSE. The formula that is applied is shown in equation (6).

$$NMSE = \frac{\text{norm}(\text{orgSig} - \text{recSig})^2}{\text{norm}(\text{orgSig})^2 \cdot \text{length}(\text{orgSig})} \quad (6)$$

The orgSig is the true data, recSig is predicted data.

D. Cross validation

In [10], it is worthy, more accurate and interpretative to apply k-folding cross validation (CV). The data is divided into k folds, one fold is for testing and k-1 folds are used to train a model. The accuracy is the average of all k iterations as explained in [10]. We have a data set of 504 values for each external factor, which is greater than 100, and apply k-fold CV for the training data.

III. EXTERNAL FACTORS

The predictors, x_{ij} , are hourly data of external factors that affect the electricity prices. These are the following as shown in Table I, prices of fossil fuels with their units, the production of renewable energy sources in MWh, total consumption, production and net exchange in MWh. The prices of fossil fuels are only given as daily data. We take the mean value of each day's high and low value and use it as the constant value for each day. During both training and testing periods, there are a few days without data for the prices of the fossil fuels. This is solved by applying Mean/Mode Imputation (MMI) [11]. We assume in this project that the external factors are independent, resulting in independent columns in the predictor matrix X , so that it becomes full rank and follows the third restriction for lasso.

Prices of fossil fuels are chosen as one of the external factors, because there is a strong relation between electricity prices and fossil fuel prices in the industry sector as shown in [1]. The fossil fuels selected in this project are: Coal, Heating Oil, Natural Gas, Gasoline, Oil (Brent) (which is a type of oil drilled from the North Sea bordering the UK and Norway), and Oil (WTI) (which is a type of oil mined in the US, according to Investopedia) [12]. Electricity is mainly produced by renewable sources as shown in Fig. 2 [3]. Besides, in the article [2], the authors find that there is a relationship between electricity prices and hydro power. Therefore, hydro power along with two more renewable resources i.e., nuclear power, wind power are selected as three of the external factors.

The data for prices of fossil fuels (Coal, Heating Oil, Natural Gas, Gasoline, Oil(Brent) and Oil(WTI)) are derived from MARKETS INSIDER [13]. The prices are daily prices taking into account the dollar/SEK exchange rate but are completed into hourly prices, as explained in Subsection III. Data for Swedish net exchange of energy is per hour with unit MWh, and it is collected from Nord Pool [14]. Swedish production of energy and Swedish consumption of energy have unit MWh and are hourly data derived from Nord Pool as well [15] - [16]. Hourly data for the production of Swedish hydro power, nuclear power and wind power are converted to MWh to keep the unit consistent [17].

A. Detecting change of dominating external factors

The project aims to compare the performance of lasso when applying it to make three different parametric models of hourly electricity prices as functions of its external factors described in Table I. The three parametric models are for different time periods of the day. The first time period spans



Fig. 2. Electricity production in Sweden 2021, the picture is from Energimyndigheten [3]

TABLE I
AFFECTING EXTERNAL FACTORS WITH CORRESPONDING UNITS.

External Factors	Unit
Coal	SEK/Ton
Heating Oil	SEK/Barrel
Natural Gas	SEK/MMBtu
Gasoline	SEK/Gallone
Oil (Brent)	SEK/100 Liter
Oil (WTI)	SEK/100 Liter
Hydro Power	MWh
Nuclear Power	MWh
Wind Power	MWh
Net Exchange of Energy	MWh
Production of Energy	MWh
Consumption of Energy	MWh

over 24 hours, meaning the whole day. The other two are two-hour long periods from 8 to 10 o'clock and 16 to 18 o'clock respectively. The model parameters in $\hat{\beta}_1$ for each hour in the day are estimated in MATLAB, and then observed and analyzed. Fig. 3 shows a linear diagram with the absolute values of the model parameters in $\hat{\beta}_1$, plotted for each hour of the day. The index of the model parameters with the highest and lowest absolute values changes before and after the time periods 8 to 10 o'clock and 16 to 18 o'clock, and are consistent throughout these periods. These periods are interpreted as periods where the parametric model of the whole day changes.

As explained in II-A, each of the model parameters in β_1 are multiplied with one column each in the predictor matrix X when performing the matrix multiplication $X\beta_1$ in the linear model. The data in each column of X comes from a separate external factor. This means that when multiplying the model parameters in β_1 that have the largest and smallest absolute values with its corresponding external factor in the matrix multiplication, these external factors become, respectively, the most significant and insignificant for the parametric model.

In Fig. 4, each one-hour period of the day, from the training data which consists of 21 days, is plotted so that there are 21 consecutive data values for each one-hour period.

This means that the diagram starts with 21 values of the electricity price from the hour 00:00 until 01:00 and ends with 21 values of the electricity price from the hour 23:00 until 00:00. The time periods 8 to 10 o'clock and 16 to 18 o'clock are marked in the figure with red and green respectively, to visualize how the underlying pattern for the electricity prices changes for these periods.

IV. IMPLEMENTATION

The project is conducted mainly in MATLAB, using built in functions for lasso and NMSE, with data management in Excel. In order to obtain the parametric models, we insert required imported data and lambda into the lasso function, which can be seen in Algorithm 3. Normalization of data is achieved by a for-loop which firstly subtracts a minimum value and thereafter divides by the difference between the maximum and minimum values, which is shown in Algorithm 1. In the for-loop shown in Algorithm 2, 20 values of lambda are obtained. The regularization parameter starts at five and decreases by 75% of the previous index, repeatedly 19 times. Further, in Algorithm 3, we use 2-fold cross validation with respect to λ , by obtaining an equal amount of values i.e., 20, for each unknown coefficient (β_1 and β_0). Out of the 20 values for λ , β_1 and β_0 , the value with the lowest MSE is used in the parametric models for which we insert the test data. The number of folds in the cross-validation is chosen by observing the number that gives the best results in various tests.

The time periods for the data sets for training and testing are presented in Table II. These are the same for each of the three parametric models and predictions. The time periods for the data of the regressors and the regressand are shifted with eight days.

Algorithm 1 Algorithm for normalisation

```

n ← sample size
x1 ← regressors
for <i = 1 to n> do
    <x1 = x1 - min(x1) ;>
    <x1 = x1 / (max(x1) - min(x1)) ;>
end for

```

Algorithm 2 Algorithm for obtaining values of λ

```

λ(1) ← 20
for <i = 2 to 20> do
    <λ(i) = λ(i-1)*0.75 ;>
end for

```

V. RESULTS & DISCUSSION

A. Different time periods

As explained in Section I, the project focuses on comparing three parametric models from different time periods. The first parametric model spans throughout the whole day, the other two periods are shorter and detected from this one. The method

Algorithm 3 Training algorithm

Require: Independence between regressors, linear relationship between regressors and regressand
Input: $x_1 \leftarrow$ regressors, $y_1 \leftarrow$ regressand, λ
Output: *parametric model*
 $[\hat{\beta}_1, \hat{\beta}_0] = \text{lasso}(x_1, y_1, 'CV', 2, 'Lambda', \lambda);$
parametric model = $X \cdot \hat{\beta}_1 + \hat{\beta}_0;$
return parametric model

TABLE II
TIME PERIODS FOR THE TRAINING DATA AND TEST DATA FOR BOTH THE REGRESSORS AND THE REGRESSAND

Type of data	Training data	Test data
Regressand	17 th of Feb – 9 th of Mar	10 th of Mar – 17 th of Mar (predicted values)
Regressors	9 th of Feb – 1 th of Mar	2 nd of Mar – 9 th of Mar

that is used is explained with more details in Section III-A and the result is illustrated in Fig. 3, obtaining the time periods 8 to 10 o'clock and 16 to 18 o'clock for the additional parametric models. The two shorter time periods are marked in red (8 to 10 o'clock) and green (16 to 18 o'clock) in Fig. 4 that shows the test data.

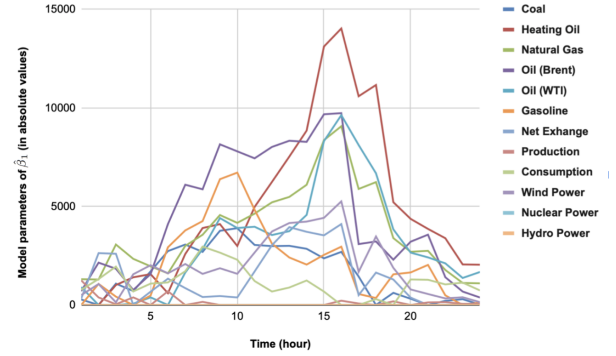


Fig. 3. Linear diagram of the model parameters of β_1 in absolute values

B. The parametric model for the whole day

The result of the prediction with the data in Table II can be seen in Fig. 5. The orange curve represents the predicted data and the blue curve represents the true data of the electrical energy prices. As shown in Fig. 5, the prediction for the whole day roughly follows the true values with approximately three times bigger peaks. The result might be due to that the applied data does not meet the assumptions for lasso. For example, correlation between the predictors and responses might not be linear enough, and there might be some unexpected dependence between the external factors in the predictor matrix, which can be investigated in further studies.

The calculated NMSE value is approximately 0.0162. This is the lowest value compared to the other two models,

meaning highest statistical accuracy in this regard, which can be seen in Fig. 8. A reason for this could be that the whole day consists of many shorter time periods, and several of these might have a relatively high linear correlation between input and output data. The combination of these might yield into a higher statistical accuracy than for the predictions of the shorter time periods with higher NMSE.

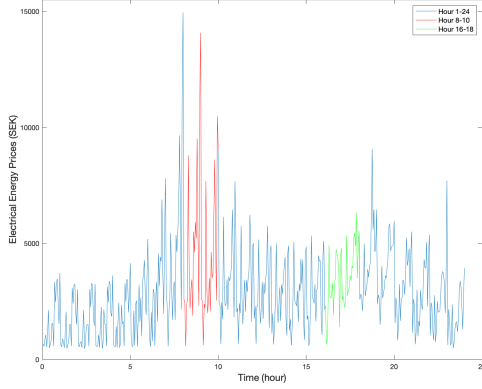


Fig. 4. Electrical energy prices with respect to 24 consecutive time periods, one for each daily hour. The data is the same as the training data, consisting of 21 days.

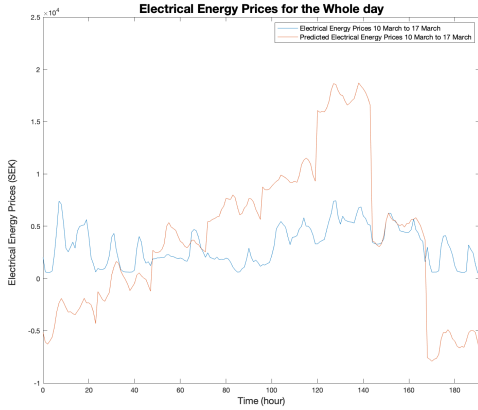


Fig. 5. Hourly prediction of electrical energy prices between 10th of March and 17th of March for the whole day

C. The parametric model for the time period 8 - 10 o'clock

Fig. 6 shows the result of the prediction with the data in Table II, compared to the true data. The orange curve represents the predicted data and the blue curve represents the true data of the electrical energy prices. The predicted and the real values follow a similar trend regarding how the prices change, considering when the first and second peak occurs. But the prediction gives approximately three times larger value at the peak and a NMSE value which can be observed in Fig. 8. This value is approximately 0.128 and the largest out of the three models. A reason might be that this period has a high uncertainty, due to being a peak period. As can be seen in the training data, marked with red in Fig. 4, the values are quite similar for some consecutive days and change drastically for other days, making it difficult to predict accurate values.

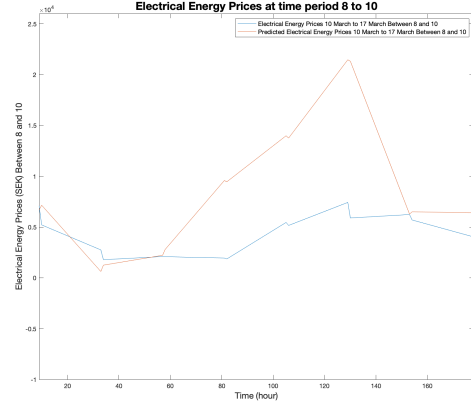


Fig. 6. Hourly prediction of electrical energy prices between 10th of March and 17th of March for the time period 8 to 10 o'clock

D. The parametric model for the time period 16 - 18 o'clock

The results of the prediction compared to the true values can be seen in Fig. 7, the training and test data is written in Table II. The orange curve represents the predicted data and the blue curve represents the true data of the electrical energy prices. This prediction fits the real data quite well, though the prediction is not sufficiently accurate. Because even though the predicted and real values follow the same trend, two of the peaks give approximately 49 % higher values for the predicted compared to the real values. As mentioned about the prediction for the whole day, a reason might be that the applied data does not meet the requirements for lasso.

The NMSE value is approximately 0.0197, as can be seen in Fig. 8, which gives the second smallest NMSE out of the three predictions. Even though the applied data does not give perfect predictions with the lasso method, the prediction gives approximately 6.53 times lower NMSE value compared to the prediction for the time period for 8 - 10 o'clock. This could mean that the correlation between the predictors and responses are more linear for this time period in particular, and therefore suits better with lasso.

E. Observing key external factors

As shown in Table III, during all three time periods, the most influential external factor is heating oil. A potential reason could be that the weather in Sweden is cold for most of the year, which leads to a great demand for thermal power for which heating oil is a source. Therefore, the price of heating oil increases when the need of it increases, the price of electricity might be influenced to a large extent because electricity can be generated by the water steam from burning the oil [18].

The time periods whole day and 16 to 18 o'clock have oil (brent) as the second most significant external factor. This fuel is used for multiple purposes, for instance, it can be used as fuel for transportation such as cars, flights, trains etc. A possible reason is that there are other alternatives than oil (brent) in developed countries such as Sweden. For example

with fuels for cars, while oil prices increase, consumers prefer alternatives such as electrical fuel for their cars, which could be a potential reason why electricity prices are influenced by the prices of this fuel. Sweden imports oil mostly from the North Sea, where this type of oil comes from, while oil (WTI) is imported from the US [12]. This could be another potential reason for why oil (brent) has a bigger impact on Swedish electricity prices than oil (WTI). The second most

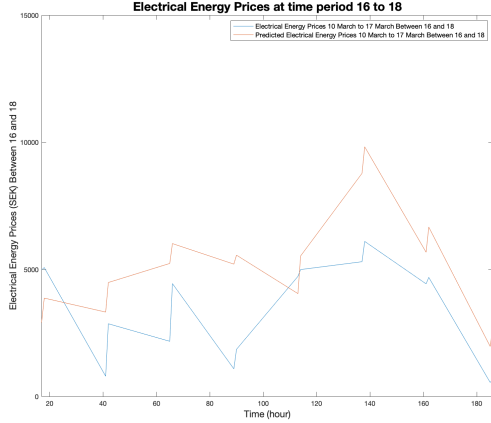


Fig. 7. Predicted Electrical Energy Prices between 10th of March and 17th of March for the time period 16 to 18 o'clock

significant external factor for the time period 8 to 10 o'clock is net exchange. This is a peak period where the demand is very high and if it is higher than the supply in an area, it is dependent on imported electricity from other areas. This might be a reason that net exchange is the second most significant external factor during this period.

F. Uncertainty of Cross Validation

Cross Validation divides the data into different sets of training data and test data every time the code is run on MATLAB, which leads to varying values of the coefficients i.e., varying results of predictions. Even though the variation is not big, it still causes a certain degree of uncertainty.

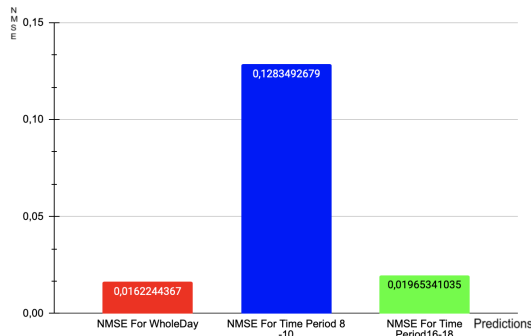


Fig. 8. NMSE for different predictions based on different time periods

G. Future Work

In future work, we would like to make research about and experiment with more external factors that affect electricity

TABLE III
THE TWO MOST SIGNIFICANT EXTERNAL FACTORS DURING DIFFERENT TIME PERIODS

Time Period	Most Significant External Factor	Second Most Significant External Factor
Whole Day	Heating Oil	Oil (Brent)
8-10 o'clock	Heating Oil	Net Exchange
16-18 o'clock	Heating Oil	Oil (Brent)

prices, since, as mentioned in Section V-B, the correlation between the regressors and regressand in this project might not have been linear enough to be applicable for lasso. Furthermore, the independence of each regressor could be further investigated, to make sure that it meets the requirements for lasso described in Section II-A. Further work can also focus on using other methods than CV in order to obtain λ and avoid varying results. In order to obtain more accurate predictions, another suggestion for future work is to conduct the predictions by applying nonlinear models since the relationship between different external factors are potentially nonlinear as explained in Section V-B.

VI. CONCLUSION

In this project, lasso is applied to predict electricity prices using three different models with the consideration of underlying affecting factors. The models are based on different time periods of the day, the whole day, 8 to 10 o'clock and 16 to 18 o'clock. Based on the identified models, we investigate the prediction performances in terms of normalized mean square error and identify the most important factors that affect the electricity price during each time period. The most accurate prediction that gives the smallest NMSE is the model of the whole day. However, even the most accurate one does not provide an adequate accuracy of prediction and a reason might be that the applied data does not meet the assumptions for lasso.

ACKNOWLEDGMENT

The authors would like to thank the supervisor Yu Wang for patient guidance and great support.

REFERENCES

- [1] M. Madaleno, V. Moutinho, and J. Mota, "Long and short-run relationship among electricity and fossil fuel prices in the european industry sector," in *2015 12th International Conference on the European Energy Market (EEM)*, 2015, pp. 1–6.
- [2] K. Laitinen, J. Hovila, T. Mannila, and L. Korpinen, "The influences of climatic factors on electricity prices in liberalized market in finland," in *DRPT2000. International Conference on Electric Utility Deregulation and Restructuring and Power Technologies. Proceedings (Cat. No.00EX382)*, 2000, pp. 544–548.
- [3] Energimyndigheten. (2022, Feb.) Energimyndigheten. Energimyndigheten, Eskilstuna, Sweden. elproduktion. [Online]. Available: <https://www.energimyndigheten.se/nyhetsarkiv/2022/fortsatt-hog-elproduktion-och-elexport-under-2021/>
- [4] D. Chen and D. W. Bunn, "Analysis of the nonlinear response of electricity prices to fundamental and strategic factors," *IEEE Transactions on Power Systems*, vol. 25, no. 2, pp. 595–606, 2010.

- [5] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996. [Online]. Available: <http://www.jstor.org/stable/2346178>
- [6] P. Händel, "Understanding normalized mean squared error in power amplifier linearization," *IEEE Microwave and Wireless Components Letters*, vol. 28, no. 11, pp. 1047–1049, 2018.
- [7] D. L. Mohr, W. J. Wilson, and R. J. Freund, "Chapter 7 - linear regression," in *Statistical Methods (Fourth Edition)*, fourth edition ed. Academic Press, 2022, pp. 301–349. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780128230435000072>
- [8] Ryan Tibshirani. (2017, Mar.) Sparsity, the Lasso, and Friends. Carnegie Mellon University, Pittsburgh, Pennsylvania, USA. Lecture notes in course Machine Learning 10-702. [Online]. Available: <https://www.stat.cmu.edu/~ryantibs/statml/lectures/sparsity.pdf>
- [9] T. Hastie, R. Tibshirani, and J. Friedman, "Elements of statistical learning: Data mining, inference, and prediction," ser. Springer series in statistics. New York: Springer, 2009, p. 68.
- [10] S. Yadav and S. Shukla, "Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification," in *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, 2016, pp. 78–83.
- [11] M. Gimpy, "Missing value imputation in multi attribute data set," *Int J Comput Sci Inf Technol*, vol. 5, no. 4, pp. 1–7, 2014.
- [12] The investopedia team. (2021, Aug.) Brent crude vs. west texas intermediate: The differences. Investopedia, New York, USA and Edmonton, Canada. Brent VS WTI. [Online]. Available: <https://www.investopedia.com/ask/answers/052615/what-difference-between-brent-crude-and-west-texas-intermediate.asp>
- [13] Business Insider. (2022, Mar.) Markets insider. Business Insider, New York, NY, USA. Prices of fossil fuels. [Online]. Available: <https://markets.businessinsider.com>
- [14] Nord Pool. (2022, Mar.) Data for net exchange. Nord Pool AS, Lysaker, Norway. [Online]. Available: <https://www.nordpoolgroup.com/en/Market-data1/Power-system-data/Exchange1/SE/Hourly2/?view=table>
- [15] NordPool. (2022, Mar.) Data for production. Nord Pool AS, Lysaker, Norway. [Online]. Available: <https://www.nordpoolgroup.com/en/Market-data1/Power-system-data/Production1/Production1/ALL1/Hourly1/?view=table>
- [16] Nord pool. (2022, Mar.) Data for consumption. Nord Pool AS, Lysaker, Norway. [Online]. Available: <https://www.nordpoolgroup.com/en/Market-data1/Power-system-data/Consumption1/Consumption/ALL/Hourly1/?view=table>
- [17] Mimer. (2022, Mar.) Data for hydro power, nuclear power and wind power. Svenska kraftnät, Sundbyberg, Sweden. [Online]. Available: <https://mimer.svk.se/ProductionConsumption/ProductionIndex>
- [18] World Nuclear Association. (2022, May) Where does our electricity come from? World Nuclear Association, London, United Kingdom. [Online]. Available: <https://world-nuclear.org/nuclear-essentials/where-does-our-electricity-come-from.aspx>

Short Term Stock Price Prediction Using Machine Learning

Alexander Wikström, Olov Rahm

Abstract—This report assesses different machine learning models' accuracies to predict whether a stock will go up or down in value in a short term. The models that is used is linear regression, LSTM and Elman RNN. These models was trained on historical price data from the Nasdaq Stock Exchange. The idea that there exist a relationship of the price movement of a stock and its future value is called 'technical analysis'. The result shows that neither LSTM nor Elman RNN provides any statistical significance of its accuracy for any of the implementations. Linear regression, provides a significant accuracy for longer time series prediction of the price when trained on 100 days of data and prediction of its movement after five more days.

Sammanfattning—I denna rapport undersöks olika maskininlärningsmodellers noggrannhet för att förutspå om en aktie kommer att gå upp eller ner i värde på kort sikt. De evaluerade maskininlärningsmodellerna är följande: linjär regression, LSTM och Elman RNN. Dessa modeller tränades med hjälp av historisk prisdata från Nasdaq Stock Exchange. Idéen om att det finns ett samband mellan prisrörelsen av en aktie och dess kortsiktiga framtida värde är benämnt som 'teknisk analys'. Resultaten visar att varken LSTM eller Elman RNN förmedlar en noggrannhet med statistisk signifikans för någon av de använda implementationerna. Linjär regression förmedlar en statistisk signifikant noggrannhet för längre tidserie förutsägelser med träningsdata om 100 dagar och förutsägelse av aktiens rörelse efter fem fler dagar.

Index Terms—Machine Learning, Long Short-Term Memory, Recurrent Neural Network, Stock Price Prediction, NASDAQ

Supervisors: *Rebecka Winqvist*

TRITA number: *TRITA-EECS-EX-2022:132*

I. INTRODUCTION

"In the short run, the market is a voting machine but in the long run, it is a weighing machine" writes Benjamin Graham in his book *The Intelligent Investor* [1]. The stock market is an open market where anyone who can and wants can be involved; the idea is that through this openness of the market companies can get funding by individuals and investors for their projects. This is however, just the surface. The stock market has in the recent years become more and more complex with different financial instruments and derivatives available to buy with just a few clicks on your computer. The digitalization of the market has increased the trading volume and more and more people are trying to predict the market in the short term through high-frequency trading.

The idea of trying to predict how different stocks will move is an attractive one because of the high rewards associated with it. Some stakeholders in the stock market try to predict the future value of different stocks using solely previous time series data of the stocks prices. This is known as technical

analysis and is widely used despite the lack of empirical evidence of its performance.

Technical analysis is divergent from the efficient-market hypothesis (EMH) which is a hypothesis that states that all financial assets reflect all available information [2]. Therefore all prediction made in a short term horizon could be seen as stochastic. This means that the success of day traders are solely due to luck and hence invalidates the day traders claim of their success being due to skill.

The assumptions of EMH is the foundation of today's modern portfolio theory (MPT) built on by Harry Markowitz in 1952 [3]. The modern portfolio theory is a mathematical model that aims to find a portfolio of different stocks that maximize the expected value and minimize the volatility of the portfolio. The assumptions made in this model which is in resonance with EMH is that the risk of the asset is completely decided by the variance of the stock.

In Horne and Parkers paper "The random-walk theory: An empirical test" [4] they came up with the conclusion that the stock market in the short term could be seen as completely stochastic. Despite this there is still a lot of people that believes it is possible to predict the stock market in the short term. In this report the idea of short term prediction of the stock market will be tested through using different machine learning models.

A. Project Aim

The aim of the project is to test whether it is possible to predict whether a stock will go up or down in value through based on its historical price data. The models that will be used and assessed for these predictions will be linear regression, RNN, and LSTM.

B. Related Work

The authors would like to mention related works which excited inspiration to this report, which is H. Forslund and M. Johnson's bachelor thesis "*Machine Learning Methods for Predicting Trading Behaviour of an Actively Managed Mutual Fund*" [5] where the premise was to try to predict trading patterns in an actively managed mutual fund. This gave us the inspiration to try to assess whether it was possible to get significant accuracies of different machine learning models when using individual stocks, instead of a mutual fund.

II. BACKGROUND

A. The Capital Asset Pricing Model

The capital allocation model was introduced by William F. Sharpe in 1964 [6]. The CAPM is a simplification of MPT,

and has the following assumption that there exist a linear relationship between the expected value of a stock's price to the risk free rate and the market as a whole.

$$\mu_i = r + \beta_i(\mu_M - r) \quad (1)$$

where μ_i is the expected return of the individual stock, μ_M the expected return of the market β_i the correlation factor and r the risk free rate. This model assumes that there exist a linear relationship between the market and the individual stocks.

B. Linear Regression

The basic assumptions of a linear regression is that there exist a linear relationship between our response variable and our regressor variable. In this approach, an assumption that there exist a linear relationship between time and the stocks price is made, therefore we want to estimate the values of in the following equation:

$$y_i = \beta_0 + \beta x_i + \varepsilon_i \quad (2)$$

where ε_i the error, y_i the price of the stock, x_i the time.

For multiple dimension, and regressor variables we can rewrite Eq. (2) as the following:

$$\hat{y} = \beta X + \varepsilon \quad (3)$$

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1k} \\ 1 & x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nk} \end{bmatrix} \quad (4)$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}, \varepsilon = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix} \quad (5)$$

where notation is explained in (4) and (5). The least square estimator of $\hat{\beta}$ will then become as:

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad (6)$$

C. Artificial Neuron

Artificial neural networks are inspired by the brains architecture. The brain consists of billions of neurons wired together in a cluster, where every neuron can enter either a firing state or a resting state [7], i.e. either being active or dormant. The neuron enters the firing state if the amount of received stimuli surpasses a certain threshold, else it remains in a resting state [8]. In an artificial neural network (ANN), a neuron is modeled as a function $a : \mathbf{R}^n \rightarrow \mathbf{R}$ given by $a(\mathbf{x}) = \phi(\sigma(\mathbf{x}) - T)$ and is simply referred to as a neuron function [9]. Where $T \in \mathbf{R}$ is the neurons's *threshold*, $\sigma : \mathbf{R}^n \rightarrow \mathbf{R}$ is the *accumulation function* and $\phi : \mathbf{R} \rightarrow \mathbf{R}$ is the *activation function*.

In the common case, for which the accumulation function is linear, the neuron function simplifies to $a(\mathbf{x}) = \phi(\mathbf{w}^T \mathbf{x} - T)$

where the elements of \mathbf{w} are the (learnable) weights of the network.

The most widely used activation functions are the Rectified Linear Unit (ReLU), the Hyperbolic tangent (tanh), and the sigmoid function (σ) defined as follows:

$$\text{ReLU}(x) = \max(x, 0)$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

D. Deep feedforward network

$$\mathbf{x}^{i+1} = \begin{bmatrix} a_1^{i+1}(\mathbf{x}^i) \\ a_2^{i+1}(\mathbf{x}^i) \\ \vdots \\ a_m^{i+1}(\mathbf{x}^i) \end{bmatrix} \quad (7)$$

A deep feedforward network is one kind of artificial neural network that is of "great importance to machine learning practitioners" [10]. They are the groundwork that many modern neural networks are based upon. They utilize artificial neurons linked together in multiple layers wired one after another. A deep feedforward network consist of N layers where the first layer takes data as input while remaining layers takes the neurons from its preceding layer as input. The first layer, \mathbf{x}^1 might utilize the identity transformation function in order to mimic the input data to the artificial neural network. In that case, \mathbf{x}^1 is called the *input layer* [9]. Note that the input layer is required to consist of the same amount of neurons as dimensions of the input. As for the remaining layers, they take on the the values given by Eq. (7) where layer i consists of m neurons.

$$\mathbf{x}^{i+1} = \begin{bmatrix} \phi(\mathbf{w}_0^{i+1,T} \mathbf{x}^i - T_0^{i+1}) \\ \phi(\mathbf{w}_1^{i+1,T} \mathbf{x}^i - T_1^{i+1}) \\ \vdots \\ \phi(\mathbf{w}_m^{i+1,T} \mathbf{x}^i - T_m^{i+1}) \end{bmatrix} = \phi(\mathbf{W}^{i+1} \mathbf{x}^i + \mathbf{b}^{i+1}) \quad (8)$$

$$\mathbf{W}^{i+1} = [\mathbf{w}_0^{i+1} \mathbf{w}_1^{i+1} \dots \mathbf{w}_m^{i+1}]^T \quad (9)$$

$$\mathbf{b}^{i+1} = [-T_0^{i+1}, -T_1^{i+1}, \dots, -T_m^{i+1}]^T \quad (10)$$

$$\phi(\mathbf{x}) = [\phi(x_0), \phi(x_1), \dots, \phi(x_m)]^T \quad (11)$$

With linear accumulation functions and identical activation functions, Eq. (7) is simplified to Eq. (8) where \mathbf{W}^{i+1} , defined in Eq. (9) is the matrix containing the neurons' weights. The vector \mathbf{b}^{i+1} , defined in Eq. (10) contains the neurons' negative thresholds and is called the *bias*. In the rightmost section of Eq. (8), the activation function has been generalized to a form explained in Eq. (11) in order to allow for the simple notation.

Applying Eq. (8) iteratively results in layer \mathbf{x}^N obtaining a value either in the form of a scalar or a vector, this is interpreted as the output of the artificial neural network, hence

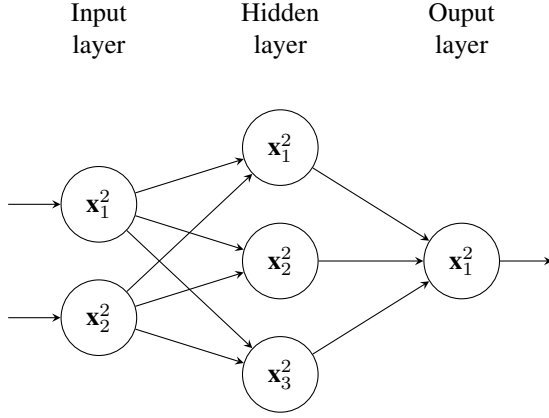


Fig. 1. An example of a deep feedforward network.

the layer \mathbf{x}^N is called the *output layer*. All layers in between the *input layer* and the *output layer* are called *hidden layers* [9].

$$\mathbf{x}^2 = \tanh(\mathbf{W}^2 \mathbf{x}^1 + \mathbf{b}^2) \quad (12)$$

$$\mathbf{x}^3 = \sigma(\mathbf{W}^3 \mathbf{x}^2 + \mathbf{b}^3) \quad (13)$$

In Fig. 1 an example of a deep feedforward network is depicted [11]. This example consists of two layers consisting of three and one neurons respectively. Data is passed to the input layer through the identity transformation function. This is further passed to the hidden layer through Eq. (12) and then further to the output layer through Eq. (13). Note that in the example network in Eqs. (12) and (13), the output layer utilizes the sigmoid function while the hidden layer utilizes the tanh function. One reason why this might be beneficial is because the two functions have their respective strengths and weaknesses. In experimental testing, using tanh function as activation function gives significantly faster convergence than using the sigmoid function [12]. One advantage with using the sigmoid function is the format of its output having the property of being interpreted probabilistic due to the range of the sigmoid function being $\sigma(x) \in [0, 1]$ [12]. However, in some problems this property is not desired and hence, other activation functions might be used in the output layer instead.

E. Training a deep feedforward network

In order for a deep feedforward network to be useful, all weights and biases in the network is adjusted to maximize accuracy. Doing so manually is manageable in smaller networks but a more efficient method is required for the task in bigger networks. The most common such method is called back-propagation. Back-propagation was first introduced in 1986 and is today vital for training neural networks [13], [14]. The founding idea of back-propagation is to minimize the "total error function" [15] with respect to the weights and biases in the network for a wide variety of inputs. Today, the "total error function" is replaced by a loss function but the target of minimizing the loss function still stands

$$MSE = \frac{1}{N} \sum_{j=1}^N |y_j - \mathbf{d}_j|^2 \quad (14)$$

One loss function called Mean Squared Error (MSE) is defined in Eq. (14) where \mathbf{y}_j is the output from the net with \mathbf{i}_j as input while \mathbf{d}_j is the desired output for said input.

$$\frac{\partial MSE}{\partial w_{1j}^3} = \frac{\partial MSE}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial w_{1j}^3} = \quad (15)$$

$$= (\mathbf{y} - \mathbf{d}) \sigma'(\mathbf{W}^3 \mathbf{x}^2 + \mathbf{b}^2) x_j^2$$

$$\frac{\partial MSE}{\partial b_1^3} = \frac{\partial MSE}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial b_1^3} = \quad (16)$$

$$= (\mathbf{y} - \mathbf{d}) \sigma'(\mathbf{W}^3 \mathbf{x}^2 + \mathbf{b}^2)$$

$$\frac{\partial MSE}{\partial w_{ij}^2} = \frac{\partial MSE}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial x_i^2} \frac{\partial x_i^2}{\partial w_{ij}^2} = \quad (17)$$

$$= (\mathbf{y} - \mathbf{d}) \sigma'(\mathbf{W}^3 \mathbf{x}^2 + \mathbf{b}^2) \tanh'(\mathbf{W}^2 \mathbf{x}^1 + \mathbf{b}^1) w_{1i}^3 x_j^1$$

$$\frac{\partial MSE}{\partial b_i^2} = \frac{\partial MSE}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial x_i^2} \frac{\partial x_i^2}{\partial b_i^2} = \quad (18)$$

$$= (\mathbf{y} - \mathbf{d}) \sigma'(\mathbf{W}^3 \mathbf{x}^2 + \mathbf{b}^2) \tanh'(\mathbf{W}^2 \mathbf{x}^1 + \mathbf{b}^1) w_{1i}^3$$

$$\Delta \mathbf{w}(t) = -\epsilon \frac{\partial MSE}{\partial \mathbf{w}}(t) + \alpha \Delta \mathbf{w}(t-1) \quad (19)$$

Note that $MSE \geq 0$ with equality if and only if $\mathbf{y}_j = \mathbf{d}_j \forall j$. With this insight in mind, the problem of maximizing the networks accuracy can be rephrased as minimizing the loss function. Minimizing the loss function using gradient descent requires the partial derivative of said loss function w.r.t. all weights and biases to be calculated. Using the deep feedforward network in Fig. 1 with one input-target pair as example, the chain rule can be utilized. Demonstrated in Eqs. (15) to (18) the calculations are made.

In the more common case of a data set of more than one data point, the gradient is set to the sum of partial derivatives [15]. For every step, the parameters are updated as $\mathbf{w} = \mathbf{w} + \Delta \mathbf{w}$ where $\Delta \mathbf{w}$ is given by Eq. (19) where $\epsilon \in \mathbf{R}$ and $\alpha \in [0, 1]$ is an *exponential decay factor* [15].

This method of back-propagating is the foundation of modern back-propagation. However, a lot of improvements and modifications has been made to the method. The original loss function isn't ideal in all examples and hence, numerous different loss function has been created since the beginning. The algorithm for updating the weights in an artificial neural network has also been improved since the implementation of Eq. (19). These algorithms are in modern terminology referred to as *optimizer algorithms* [16].

F. Vanishing gradient problem

In deep feedforward networks with many layers, a phenomenon called *vanishing gradient problem* might occur. By studying the patterns in Eqs. (15) to (18) one can conclude that gradients of parameters in early layers contains more factors corresponding to the derivative of an activation function. E.g. in a deep feedforward network with N layers, gradients

of parameters in layer i will contain $N - i + 1$ factors corresponding to derivatives of activation functions [15].

$$\tanh'(x) = 1 - \tanh^2(x) \quad (20)$$

$$\sigma'(x) = \sigma(x)(1 - \sigma(x)) \quad (21)$$

Studying the derivatives of the two activation functions $\tanh(x)$ and $\sigma(x)$ given by Eqs. (20) and (21) and depicted in Fig. 2 [15] the conclusion that $\sigma'(x) \leq 0.25$ and $\tanh'(x) \leq 1$ is made.

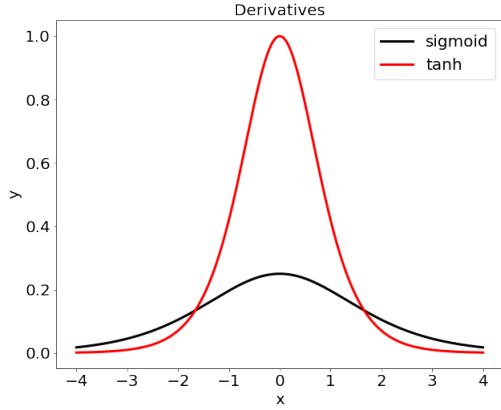


Fig. 2. The derivatives to tanh function sigmoid function.

For a deep feedforward network with many layers utilizing the sigmoid activation function, gradients of early layers are a product of (among others) multiple derivatives ≤ 0.25 . This may result in a very small gradient [17].

The problem that negligible gradients cause is that essentially, the weights are not updated and hence takes too long to train [17], [18].

G. Recurrent Neural Networks

For finding patterns in a time series, deep feedforward networks are lacking according to J.L. Elman the reason being they "do not easily distinguish temporal dimension from spacial dimensions" [19]. To overcome these shortcomings, a new type of structure called recurrent neural network (RNN) was developed. RNN utilizes a "memory" in order to represent the temporal dimension [19]. The architecture of the Elman RNN is depicted in Fig. 3, the figure is created by the authors based on the equations from PyTorch documentation [20].

An Elman RNN can be imagined as a deep feedforward network where the amount of inputs might be interpreted as the amount of layers in a deep feedforward network. Hence, the Elman RNN suffers from the vanishing gradient problem in the sense that gradients from early inputs becomes vanishingly small [21].

Due to the shear complexity of the stock market, a model that suffers from short term memory is probably lacking and not able to make predictions of adequate quality. It is advantageous for the network to be able to utilize data early in

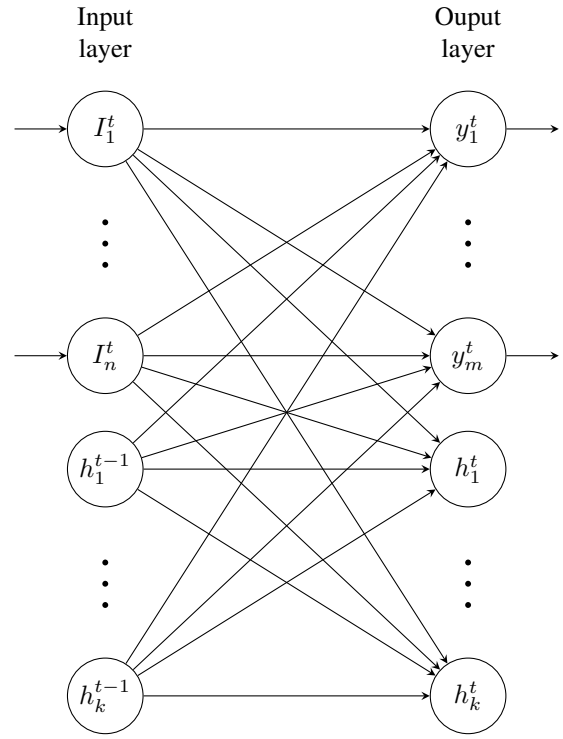


Fig. 3. A Recurrent Neural Network.

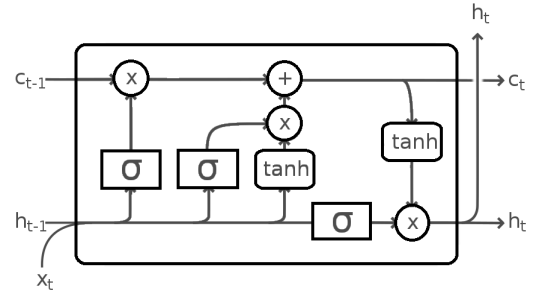


Fig. 4. An LSTM cell. [22]

order to make a prediction. Hence this model isn't sufficiently pleasing and needs to be improved further. The prediction of an Elman RNN will however be evaluated in this report.

H. Long Short Term Memory Neural Network

$$z_t = [h_{t-1}, x_t] \quad (22)$$

$$f_t = \sigma(W_f z_t + b_f) \quad (23)$$

$$i_t = \sigma(W_i z_t + b_i) \quad (24)$$

$$N_t = \tanh(W_N z_t + b_N) \quad (25)$$

$$c_t = c_{t-1} \cdot f_t + N_t i_t \quad (26)$$

$$u_t = \tanh(W_u c_t + b_u) \quad (27)$$

$$v_t = \sigma(W_v z_t + b_v) \quad (28)$$

$$h_t = U_t \cdot V_t \quad (29)$$

The main flaw of the simple RNN is as previously mentioned, the lack of long term memory. The Long Short Term Memory Neural Network (LSTM) is a type of RNN designed with focus on solving this short memory problem. The LSTM solves this problem by implementing a long term memory that is passed in parallel to the earlier mentioned hidden state that suffers from short term memory. This long term memory is called a memory cell c_t . [23]. At every time step, the memory cell is updated using the information gained from the input and hidden state.

An LTSM cell consists of four gates [24]; the forget gate (Eq. (23)), the learn gate (Eqs. (24) and (25)), the remember gate (Eq. (26)) and the use gate (Eqs. (27) to (29)) where z_t is defined in Eq. (22).

The idea behind these four gates are as follows:

- The learn gate filters non-useful information from the input and short term memory.
- The forget gate filters non-useful information from the long term memory.
- The remember gate combines the filtered long and short term memories to create a new long term memory.
- The use gate combines the filtered long and short term memories to create a new short term memory.

The LSTM architecture is visualized in Fig. 4 [24].

III. IMPLEMENTATION

A. Method

The models will be given a time series of a stock's prices i , that spans from day $T_i^0 + 1$ to day $T_i^0 + T$, i.e. a time series with T days worth of stock prices. It will then predict whether the closing price of day $T_i^0 + T + E$ is higher or lower than the closing price of day $T_i^0 + T$. Four different values of the tuple (T, E) will be tested. These are:

- $(T, E) = (20, 1)$
- $(T, E) = (20, 5)$
- $(T, E) = (100, 1)$
- $(T, E) = (100, 5)$

Note that a time series-target pair is defined uniquely from the variables (T_i^0, T, E) and the stock.

B. Choice of data

To collect data, the python library **yfinance** [25] will be used. This library provides lots of information about all the stocks in NASDAQ's stock exchange. In **yfinance**, there exist a total of seven variables in the form of time series. Those are:

- Open: The price at the start of the interval.
- High: The highest price during the interval.

- Low: The lowest price during the interval.
- Close: The price at the end of the interval.
- Volume: The amount of stocks traded during the interval.
- Dividends: The amount of dividends performed during the interval.
- Stock Splits: The amount of stock splits performed during the interval

The data used is four first mentioned variables: Open, High, Low and Close. The reason being these are all the available information with a direct connection to the stock price.

In the **yfinance** library there exists a total of 8449 different stocks for which the price history of a stock can range from 0 to over 15000 days. Out of one stock many independent time series can be gathered, for example we can create two time series-target pair defined by (T_i^0, T, E) and $(T_j^0 = T_i^0 + T, T, E)$. Despite the data point at time $T_i^0 + T + E$ appearing in both time series (assuming $E < T$), these training instances will be considered independent from one another.

The time series-target pairs will be chosen as follows:

- 1) Choose T and E and keep them fixed.
- 2) For each stock, add the time series-target pairs defined by $(T_i^0 = i * T, T = T, E = E)$ until no more independent time series-target pairs are left (i.e. while $i * T + T + E < datapoints$).
- 3) The first 80% of the pairs (rounded down) are stored as potential training pairs. Remaining pairs are stored as potential testing pairs.
- 4) a random sample of 10000 elements from the potential training data are used training data in the model.
- 5) a random sample of 10000 elements from the potential testing data are used testing data in the model.

The price history of some stocks contains price = **0** or price = **nan**. In these cases, none of the pairs from that stock is added to potential training or potential testing data.

C. Scaling

The idea when scaling the data is that only the *relative* change in the stocks price is relevant. Hence, all input-target pairs are scaled individually s.t. the opening price of the first day is one and all other prices are relative to that one price. i.e.:

- 1) Choose a input-target pair.
- 2) Set p_0 to the opening price of the first day.
- 3) Divide all prices with p_0 .

Hence, all pairs are scaled equally with only the change relative to p_0 being relevant.

D. Elman RNN hyperparameters

In order to model the RNN, the PyTorch RNN module was implemented with `hidden_size = 128`, `input_size = output_size = 4`. All other hyperparameters are default. The interested reader is referred to [20]. The four inputs/outputs are the open, close, high and low of each day.

E. Elman LSTM hyperparameters

In order to model the LSTM, the PyTorch LSTM module was implemented with $hidden_size = 128$, $input_size = output_size = 4$. All other hyperparameters are default. The interested reader is referred to [26]. The four inputs/outputs are the open, close, high and low of each day.

F. Training the machine learning models

When training the machine learning models the data points acquired by using the previously mentioned method will be shuffled and 1000 batches of size ten will be made and iterated over a total of ten epochs, i.e. all data is looped through 10 times.

G. Result evaluation

To evaluate the different models, a *confusion matrix* will be implemented. The confusion matrix is defined as

$$CM = \begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix}$$

Where TP = True Positives, FP = False Positives, FN = False Negatives, TN = True Negatives.

Given the confusion matrix of a model, the model will be evaluated by studying its $sensitivity = \frac{TP}{TP+FN}$, its $specificity = \frac{TN}{TN+FP}$, and its $accuracy = \frac{TP+TN}{TP+FP+FN+TN}$ where all three will take on a value between 0 and 1. The higher the value, the better the model.

H. Statistical Validation

To check if the model is statistically significant we will use the following hypothesis test: Let $X_j \equiv$ the accuracy for model j

$$H_0 : X_j = 0.5 \text{ The method is random}$$

$$H_1 : X_j \neq 0.5 \text{ The method is statistically significant}$$

Because $X_j \in [0, 1]$ we know that $X_j \sim \text{Bernoulli}(n, p)$, and because $n = 10000 > 40$, we can approximate

$$X_j \overset{\text{approx.}}{\sim} N(p, \frac{p(1-p)}{n})$$

To reject H_0 with a significance of $\alpha = 0.01$

$$\hat{X}_{j,obs} > p_{H_0}^* + \sqrt{\frac{p_{H_0}^*(1-p_{H_0}^*)}{n}} Z_{\alpha=0.01}, \text{ we have that}$$

$p_{H_0}^* = 0.5$ from H_0 where we want to test if the accuracy is better (or consequently worse) than 50%, $Z_{\alpha=0.01} = 2.33$ from a statistical table and $n = 10000$ so in other words. We can reject H_0 if:

$$|X_{j,obs} - 0.5| > \sqrt{\frac{0.5(1-0.5)}{10000}} * 2.58 = 0.0129$$

which will be our test statistic to validate the models accuracy, where $X_{j,obs}$ will be the observed accuracy of the model.

IV. RESULTS

A. Confusion Matrix Result

The results obtained are documented in Table I to IV where the confusion matrix, sensitivity, specificity and accuracy is shown for every implementation of the different methods.

TABLE I
CONFUSION MATRIX AND ACCURACY FOR (T, E) = (20, 1)

Method	1 day CM		Sensitivity	Specificity	Accuracy
L. R.	2517 2319	2649 2515	0.5205	0.4870	0.5032
RNN	2826 1956	3082 2136	0.5910	0.4094	0.4962
LSTM	3523 1313	3660 1504	0.7285	0.2912	0.4980

TABLE II
CONFUSION MATRIX AND ACCURACY FOR (T, E) = (20, 5)

Method	1 day CM		Sensitivity	Specificity	Accuracy
L. R.	2599 2377	2560 2464	0.5223	0.4733	0.5063
RNN	3507 1469	3471 1553	0.7048	0.3091	0.5060
LSTM	3105 1871	3067 1957	0.6234	0.3897	0.5062

TABLE III
CONFUSION MATRIX AND ACCURACY FOR (T, E) = (100, 1)

Method	1 day CM		Sensitivity	Specificity	Accuracy
L. R.	2678 2224	2723 2375	0.5463	0.4659	0.4953
RNN	2234 2668	2308 2790	0.4557	0.5473	0.5024
LSTM	3879 1023	3967 1131	0.7913	0.2219	0.5010

TABLE IV
CONFUSION MATRIX AND ACCURACY FOR (T, E) = (100, 5)

Method	1 day CM		Sensitivity	Specificity	Accuracy
L. R.	2771 2190	2624 2415	0.5586	0.4793	0.5186
RNN	3349 1612	3312 1727	0.6751	0.3427	0.5076
LSTM	1680 3281	1751 3288	0.3386	0.6525	0.4968

V. DISCUSSION

A. Interpretation of the result

The only model that fulfilled $|X_{j,obs} - 0.5| > 0.0129$ was the linear regression of $(T, E) = (100, 5)$, which had an accuracy of 0.5186. This means that we can say with a confidence of 99% that the model performed better than randomly guessing whether the stock would go up the next 5 days. The LSTM and RNN model both had an accuracy that was less than

statistically significant for all implementations, which means that no evidence was provided contradicting the hypothesis that the model is as good as just guessing whether a stock will go up or down. This illustrates that it seems to be really hard to predict the short term movement of the stock prices.

The split between the TF and TN seems to be relative even for the linear regression, which means that it predicts evenly if the stock will go up or down. The LSTM model for $(T, E) = (20, 1)$ had a sensitivity of 0.7285 which means that the algorithm predicted well when the stock was going to increase, however this got counterbalanced with the low specificity of 0.2912

B. Data bias

One thing to consider is that the data is biased for stocks that have historically performed well. This due to the fact that the data is taken from stocks that is listed on the NASDAQ stock exchange. There is thus likely to be a bias against well performing companies; because companies that has performed really bad or even went bankruptcy might not be listed today and hence not included in the data set. Another thing to consider is that this report only covers stocks from one source. The result would likely vary is an exchange from another country is chosen.

VI. CONCLUSION

The aim of this project was to see if it is possible to predict the short term value of a stock by using previous stock prices by implementing machine learning models such as a linear regression, Elman RNN and LSTM. The data that was used was all stocks currently listed on the NASDAQ stock exchange at random points in time. What could be seen was that from all of the implementations, only the linear regression for $(T, E) = (100, 5)$ had statistically significant accuracy. This gives an indicator that stocks that have been moving in a certain direction for the last 100 days will tend to keep the same track for the following 5 days. Using data from 100 days ago, however, might be rather seen as medium short term analysis, since looking at the stock market from an even longer perspective it has historically had a positive trend. The S&P-index have in average since 1957 increased by 10.2% per year [27]. This is due to macro-economical effects, such as inflation and technological development, which is captured in more long run investment decisions and not in short term.

Linear regression of the stock data is actually just a trend line, and to see whether the stock has historically gone up or down the last 20 or 100 days, one could deduce by looking at the graph. It can therefore seem useless to try to implement LSTM/RNN in the same manner. Using the linear regression as a benchmark, its reasonable to conclude that it would not be a good idea making any investments based on the models assessed in this report. This does not prove that the movement of the stocks are unpredictable, rather it gives an indication that in order to find patterns one must do more advanced method and probably use more data than just technical indicators indicators, for instance by including fundamental key performance metrics of the company.

A. Future work

Something that was not included in this model was how high the return of a stock was, which means that this model would be hard to implement because it only gives a binary output: that the stock will go up or down. This is not enough information for an investor to make a rational decision. Therefore something that could be implemented is more data outputs in the forecasting, e.g. the expected return of the stock, and include how certain the algorithm is of its prediction. One thing to include could be that the algorithm, when its unsure, could choose to classify the stock as neutral, meaning that it is unsure whether the stock will increase or decrease in value, which would likely decrease the number of wrong decisions the model will make.

ACKNOWLEDGMENT

The authors would like to thank our amazing supervisor Rebecka Winqvist for guiding us in the right direction in the complex world of machine learning. In combination with our previous knowledge and interest in the stock market and investing we believe that together with the help of Winqvist's expertise produced an interesting article.

REFERENCES

- [1] B. Graham, *The Intelligent Investor*. Prabhat Prakashan, 1965. [Online]. Available: <https://books.google.se/books?id=meDYDQAAQBAJ>
- [2] E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *The Journal of Finance*, vol. 25, no. 2, pp. 383–417, 1970. [Online]. Available: <http://www.jstor.org/stable/2325486>
- [3] 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>
- [4] J. C. V. Horne and G. G. Parker, "The random-walk theory: An empirical test," *Financial Analysts Journal*, vol. 23, no. 6, pp. 87–92, 1967. [Online]. Available: <https://doi.org/10.2469/faj.v23.n6.87>
- [5] H. Forslund and M. Johnson, "Machine learning methods for predicting trading behaviour of an actively managed mutual fund," 2021.
- [6] W. F. Sharpe, "Capital Asset Prices: A Theory Of Market Equilibrium Under Conditions Of Risk," *Journal of Finance*, vol. 19, no. 3, pp. 425–442, September 1964. [Online]. Available: <https://ideas.repec.org/a/bla/jfinan/v19y1964i3p425-442.html>
- [7] E. Harth, T. Csermely, B. Beek, and R. Lindsay, "Brain functions and neural dynamics," *Journal of Theoretical Biology*, vol. 26, no. 1, pp. 93–120, 1970. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0022519370800352>
- [8] M. Feindt and U. Kerzel, "The neurobayes neural network package," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 559, no. 1, pp. 190–194, 2006, proceedings of the X International Workshop on Advanced Computing and Analysis Techniques in Physics Research. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168900205022679>
- [9] X.-S. Zhang, *Introduction to Artificial Neural Network*. Boston, MA: Springer US, 2000, pp. 83–93. [Online]. Available: https://doi.org/10.1007/978-1-4757-3167-5_5
- [10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. [Online]. Available: <http://www.deeplearningbook.org>
- [11] D. Svozil, V. Kvasnicka, and J. Pospichal, "Introduction to multi-layer feed-forward neural networks," *Chemometrics and intelligent laboratory systems*, vol. 39, no. 1, pp. 43–62, 1997.
- [12] C. MVS, "Activation functions : Why "tanh" outperforms "logistic sigmoid"?" *medium*, Dec 2019.
- [13] Y. LeCun, D. Touresky, G. Hinton, and T. Sejnowski, "A theoretical framework for back-propagation," in *Proceedings of the 1988 connectionist models summer school*, vol. 1, 1988, pp. 21–28. [Online]. Available: https://www.researchgate.net/profile/Yann-Lecun/publication/2360531_A-Theoretical_Framework_for_Back-Propagation/links/0deec519dfa297eac1000000/A-Theoretical-Framework-for-Back-Propagation.pdf

- [14] A. Al-Masri. (2019, Jan) How does back-propagation in artificial neural networks work? [Online]. Available: <https://towardsdatascience.com/how-does-back-propagation-in-artificial-neural-networks-work-c7cad873ea7>
- [15] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *nature*, vol. 323, pp. 533–536, Aug 1943. [Online]. Available: <https://doi.org/10.1038/323533a0>
- [16] PyTorch. (2019) torch.optim. [Online]. Available: <https://pytorch.org/docs/stable/optim.html>
- [17] M. Roodschild, J. Gotay Sardiñas, and A. Will, “A new approach for the vanishing gradient problem on sigmoid activation,” *Progress in Artificial Intelligence*, vol. 9, pp. 351–360, Dec 2020. [Online]. Available: <https://doi.org/10.1007/s13748-020-00218-y>
- [18] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber *et al.*, “Gradient flow in recurrent nets: the difficulty of learning long-term dependencies,” 2001.
- [19] J. L. Elman, “Finding structure in time,” *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1402_1
- [20] PyTorch. (2019) torch.nn. [Online]. Available: <https://pytorch.org/docs/stable/generated/torch.nn.RNN.html>
- [21] A. H. Ribeiro, K. Tiels, L. A. Aguirre, and T. Schön, “Beyond exploding and vanishing gradients: analysing rnn training using attractors and smoothness,” in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, S. Chiappa and R. Calandra, Eds., vol. 108. PMLR, 26–28 Aug 2020, pp. 2370–2380. [Online]. Available: <https://proceedings.mlr.press/v108/ribeiro20a.html>
- [22] Wikipedia. (2018, aug) Long short-term memory. [Online]. Available: https://en.wikipedia.org/wiki/Long_short-term_memory
- [23] S. Hochreiter and J. Schmidhuber, “Lstm can solve hard long time lag problems,” in *Advances in Neural Information Processing Systems*, M. Mozer, M. Jordan, and T. Petsche, Eds., vol. 9. MIT Press, 1996. [Online]. Available: <https://proceedings.neurips.cc/paper/1996/file/a4d2f0d23dcc84ce983ff9157f8b7f88-Paper.pdf>
- [24] G. Singh. (2021, Jan) Understanding architecture of lstm. [Online]. Available: <https://www.analyticsvidhya.com/blog/2021/01/understanding-architecture-of-lstm/>
- [25] ranaroussi. (2022, Jan) yfinance repository. [Online]. Available: <https://github.com/ranaroussi/yfinance>
- [26] PyTorch. (2019) torch.lstm. [Online]. Available: <https://pytorch.org/docs/stable/generated/torch.nn.LSTM.html>
- [27] J. Maverick. (2022, Feb) What is the average annual return for the samp;p 500? [Online]. Available: <https://www.investopedia.com/ask/answers/042415/what-average-annual-return-sp-500.asp>

Deep Learning Methods for Recovering Trading Strategies

Oliver Spjuth and Erik Emtell

Abstract—The aim of this paper is first of all to determine whether deep learning methods can recover trading strategies based on historical price and volume data, with scarcity of real data in mind. The second aim is to evaluate the methods to generate a deep learning blueprint for strategy extraction. Trading strategies can be built on many different types of data, often combined from different areas. In this paper, we focus on trading strategies based solely on historical price and volume data to limit the scope of the problem. Combinations of different deep learning architectures and methods such as transfer- and ensemble methods were evaluated. The results clearly show that deep learning models can recover relatively complex trading strategies to some extent. Models leveraging transfer learning outperform other models when data is scarce and ensemble methods elevate performance in certain regards.

Sammanfattning—Målet med denna rapport är i första hand att ta reda på om djupinlärningsmetoder kan återskapa handelsstrategier baserat på historiska priser och volymdata, med vetskapen att datan är begränsad. Det andra målet är att utvärdera metoder för att skapa en djupinlärningsmall för att utvinna handelsstrategier. Handelsstrategier kan vara byggda på många olika datatyper, ofta i kombination från olika områden. I denna rapport fokuserar vi på strategier som enbart är baserade på historiska priser och volymdata för att begränsa problemet. Kombinationer av olika djupinlärningsarkitekturer tillsammans med metoder som till exempel överföringsinlärning och ensembleinlärning utvärderades. Resultaten visar tydligt att djupinlärningsmodeller kan återskapa relativt komplexa handelsstrategier. Modeller som utnyttjade överföringsinlärning presterade bättre än andra modeller när datan var begränsad och ensembleinlärning ökade prestandan ytterligare i vissa sammanhang.

Index Terms—Deep Learning, Recurrent Neural Network, Convolutional Neural Network, WaveNet, Ensemble Methods, Stacking, Bagging, Transfer Learning, Algorithmic Trading.

Supervisor: Javad Parsa

TRITA number: TRITA-EECS-EX-2022:133

I. INTRODUCTION

In the world of finance, the primary goal is, just as in any other industry, to increase profit. When it comes to financial markets, there are many players. In this context, the players are represented by funds, individual traders, and so on. They are all occupying the same field and have the same goal in mind. The search for a so-called “edge” in these markets is therefore of utmost importance in order to survive and thrive instead of getting trampled. An “edge” represents knowledge of a sort of market inefficiency [1], which can be utilized to increase profits.

The ability to extract information from another market participant’s trading pattern is therefore a way to gain an

“edge” or remove one from another participant by exploiting this information. This project aims to develop methods to do just this for a specific type of strategy.

Just as there are different types of players in sports, representing a defensive or offensive style of playing, there are different types of styles that these funds and traders take on. A relatively recent style, seen from a broader historical lens, is the algorithmic approach. This style comprises all types of strategies that are automated by computers [2]. It can be regarded as one that seeks to gain an “edge” in speed, as the ability to act quickly on new information is essential to create profit from unaware adversaries on the market. Market participants acting as adversaries is due to the fact that profit is gained from an “edge” in information momentarily relative to other participants rather than overall market movements. This is often correlated with the underlying economic development. However, the algorithmic style does not only consist of strategies relying on speed, but also of strategies meant to extract new unknown information that could lead to an “edge”.

The algorithmic approach is to leverage data of all kinds in order to gain predictive power, often from extensive research [3]. These models, depending on their purpose, can rely on macroeconomics, insider trading, sentiment analysis, and purely price and volume data of an instrument [3].

The strategies described above differ a lot from classical portfolio theory, such as modern portfolio theory (MPT) [4]. MPT is essentially a hypothesis on how the optimal portfolio should be created, based on the expected value, variance, and covariance among a basket of instruments. Even though the theory is based on relatively sound principles regarding the creation of a portfolio, it has become outdated in some sense [5] while algorithmic trading is on the rise [2].

When it comes to reverse engineering a trading strategy with the intent of extracting information from an adversary’s trading patterns, there are a lot of ways to approach the problem. If one assumes that an adversary is using some form of MPT to select assets for their portfolio, the problem turns into reversing the optimization problem behind MPT [6]. To assume that MPT is behind the selection of assets in today’s markets is starting to lose its footing. Although the assumption of MPT is not as sound as it once was, it is a feasible problem seen from a reverse engineering perspective.

With feasibility in mind, this project aims to develop methods to uncover trading strategies of an adversary, with the assumption that the strategies are of an algorithmic type. To reach some level of feasibility, the problem is constrained to strategies that deal only with data concerning an asset’s historical price and volume. The methods that this thesis

applies in order to solve this problem are contained inside the domain of machine learning, more specifically deep learning.

The thesis is organized as follows. Section II explains the relevant topics needed to understand the method and the experiments. Sections III and IV define the problem and the method used to solve it. Section V contains the results of the experiments and sections VI and VII discuss these and conclude the report.

II. BACKGROUND

Subsections II-A to II-E go through and explain all types of models and architectures used in this project. After this, subsections II-F to II-I contains information about more granular topics in the field of machine learning. Subsections II-J and II-K explain what binary classification is and how to measure the performance of the model. Deep learning methods employed in this project are then gone through in subsections II-L to II-O. The last subsections II-P to II-R introduce information regarding data and data generation.

A. Artificial Neural Networks

Artificial Neural Networks (ANNs) are the basis for deep learning. They were first introduced as far back as the 1940's [7] but widespread adaption first took place in the 21st century [7]. In the beginning, ANNs were seen as a representation of the brain, but they have since diverged, using ideas from a wide range of fields. The fundamental building blocks of ANNs are often called *neurons*, representing a sort of processing unit. Combining multiple neurons gives what is called a *layer*. If an ANN consists of multiple layers, the ones in the middle are referred to as *hidden layers*. Each neuron gets fed data either from the neurons in the previous layers, or raw data if the neuron is on the first layer. The neuron then transforms this input data by multiplying it by a *weight* and adding what is known as a *bias*. The optimal choice of these weights and biases is what the optimization algorithm is searching for, as seen in subsection II-G. How the weights and biases change is decided by simple derivation of the *loss function* with regards to each weight and bias, seen in subsection II-I. The process of updating the bias and weights is often done by an algorithm called *back-propagation* [7]. For the ANN to be able to learn functions that are non-linear, the ANN itself must consist of non-linear functions, the non-linear functions of an ANN are often referred to as *activation functions* [7].

Today there are many different types of ANN architecture, the most simple is called a *feedforward* neural network and it adheres to the description in the previous paragraphs [8]. Recurrent Neural Networks (RNNs), as the name implies, introduce recurrence in the network in order to make use of previous data. This was important for predicting sequential data, such as time series. Convolutional Neural Networks (CNNs) were developed in order to process and learn information from data such as images in a much more fitting and effective way but have since been applied successfully to other types of data as well [7].

B. Convolutional Neural Networks

As described in subsection II-A, Convolutional Neural Networks were first introduced with data such as images in mind. One of the most impactful papers on Convolutional Neural Networks came out as early as 1989 [9] demonstrating CNNs ability to recognize handwritten images. CNNs can be seen as a matrix if processing images and as a vector if processing a time series. Each element in the matrix or vector can then be seen as a weight. In the case of a matrix and processing of an image, the matrix can be described as "sliding" over each pixel of the image, the pixel representing the center, replacing the pixel value with a value calculated by multiplying the weight of each element in the matrix to the corresponding pixel of the image. A sort of weighted sum of n surrounding pixels [7].

The parameter known as *kernel* represents the size of these matrices or vectors, similar to the number of inputs to a simple neuron. *Filters* denotes how many different matrices should be used, and an analogy to this would be the number of neurons in a simple feedforward neural network. *Dilation* is a parameter that describes the distance between each element of the matrix or vector, essentially how far apart the inputs are. An example when dealing with a time series would be, given a dilation of 2 and kernel-size 2, the resulting vector can be seen as having 3 elements where the element in the middle is set to be constantly 0, thus skipping the middle element in calculations.

C. WaveNet Architecture

WaveNet is the name of a special type of CNN architecture that stacks convolution layers and doubles the dilation rate. The kernel size is set to two, meaning two inputs for each layer. By stacking the layers it means that the first layer gets access to two time steps while the next layer gets access to four, twice as many. This pattern continues for each consecutive layer, as seen in Figure 1. The lower layers thus learn short-term patterns while the higher layers learn long-term patterns. Doubling the dilation rate and a fixed kernel size of two allow the network to process large amounts of data efficiently while learning a wide range of patterns [10].

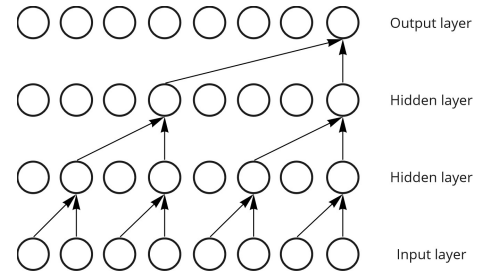


Fig. 1. Visual representation of the WaveNet architecture.

D. Recurrent Neural Networks

Recurrent Neural Networks (RNNs) were, as described in subsection II-A created with the intent of handling data taking the form of sequences in a more specialized way. To do this, a sort of memory state was introduced. In normal feedforward

neural networks, no information from the previous calculations of a neuron is saved. This is not the case for RNNs. In the simplest case, a *hidden state* [7] is introduced. This hidden state is calculated via a similar process to how outputs of normal neurons are, via *weights*. The value of the hidden state at time $t+1$ (or data point $t+1$) is calculated by combining the value of the hidden state at time t and the input to the neuron at time t . The output of a neuron in an RNN layer is then calculated by combining the hidden state and the "regular" output.

There are different ways to train and build an RNN. Since the data is sequential, an output could be given for each time step, and a loss could be calculated by comparing with a label for each step. This type of RNN network is sometimes called *sequence-to-sequence* [11]. Another type of RNN network is commonly known by the name *sequence-to-vector* [11] or *many-to-one*. As the name suggests, these networks take sequences as input, but only output a prediction after the final part of the input sequence [7].

A special kind of RNN architecture is called Long Short-Term Memory (LSTM), which was introduced in 1997 [12]. LSTMs were created to handle sequences more effectively than the classical RNN structure. Since their introduction, they have produced impressive results in a number of fields [13] and are still widely used today. To achieve this, the LSTM architecture introduces many new properties, such as input, forget, and output gates together with an internal loop [7]. A complete LSTM "neuron" can no longer be compared to a neuron in a feedforward neural network, but more as a complex network in itself.

E. Logistic Regression

Logistic regression is a type of machine learning model based on regression that is able to do binary classification. Logistic regression is essentially regression, but the input is transformed further in order to achieve an output span of $[0, 1]$. This is done so that the output can represent a probability. The model utilizes the sigmoid function; see Equation (1), also called the logistic function [14], to achieve the desired output span.

F. Activation Functions

Activation functions are functions that are employed to transform the output of neurons in the artificial neural networks. For most problems a non-linear activation function is preferred as it enables the algorithm to differentiate and classify the outputs between a set range and model complex function spaces. Four popular ones are sigmoid, tanh, ReLU, and leaky ReLU [15], which all have different properties and therefore different areas where they are most useful.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

The sigmoid function, as seen in Equation (1), is generally used to predict the output of the final layer of a binary classification task, as it returns a value between 0 and 1. The

output can thus represent a probability of the sample belonging to class 1 or 0.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2)$$

Tanh, also called hyperbolic tangent, is similar to sigmoid activation but returns a value between -1 and 1.

ReLU which was inspired from how neurons in the brain "fire", is the popular choice today when it comes to calculations in the hidden layers as it is more efficient than the tanh and sigmoid functions. The improved efficiency comes from the calculations being less complex. However, it comes with a potential problem called *Dying ReLU* phenomena described in [16]. This phenomena comes from the fact that ReLU outputs 0 if the input is negative meaning the gradient for these values are also 0, thus if the inputs are always negative the ReLU is essentially "dead". This is where the Leaky ReLU comes in.

$$\text{ReLU}(x) = \max(0, x) \quad (3)$$

Leaky ReLU combats this problem by instead of returning zero when the inputs are negative, the inputs are transformed to be very small. Therefore the gradient will not become zero and potentially "revive" the affected neuron.

$$\text{LeakyReLU}(x) = \begin{cases} x, & \text{if } x > 0. \\ 0.01x & \text{otherwise.} \end{cases} \quad (4)$$

G. SGD, the Adam Optimizer and Minimums

The Adam optimizer is a popular algorithm for training ANNs which is based upon Stochastic Gradient Descent (SGD). SGD was introduced to the field of deep learning to speed up training, since the complexity of the models and the data needed started to increase exponentially. Instead of relying on the classical Gradient Descent, which needs to evaluate the model performance on each sample before changing the weights and biases of the model, SGD only needs to evaluate a subset of the training data, also called a *batch*. This comes from a simple probabilistic assumption. The model should get an reasonable understanding of the training data, even if it is only trained on a smaller subset before changing the values of weights and biases [14]. What the model lacks in understanding from only updating on a small subset is made up by updating the weights much more often, correcting previous mistakes. The term *learning rate* refers to how large the updates of the parameters should be. The Adam optimizer improves standard SGD by using optimization techniques known as RMSprop and momentum, combining them in a highly efficient way [17]. Epoch is a term representing the number of batches corresponding to the entire data set. A model trained on 1 epoch essentially means that it has been trained on the entire data set once.

An important factor to consider with respect to optimization of ANNs is that the problem is usually *non-convex*, which means that it is not sure that the optimum reached is *global*. Furthermore, since deep learning is usually applied to complex problems, the probability of reaching the global minimum is low, since there are many *local* minimums. When training

ANNs, this means that models are very likely to reach different local minimums if they are not the same or trained on slightly different data. Two identical models trained on the same data could reach different local minimums, due to the stochastic nature of the optimizer. The solution space can be visualized as a mountain range and the local minimums would here represent the bottom of valleys. The optimizer can be seen as a explorer lost in the mountains trying to reach the sea, since sea level can be regarded as a global minimum.

H. Regularization

Regularization is a method to reduce what is called *overfitting*. Overfitting is a phenomenon characterized by a model performing better on the training data than on the validation data [7], [14]. If a model is overfitting, this can be an indication that the model is not able to generalize to new data. The ability of the model to generalize is essential for developing models that can actually be used, since generalization is a measure of how well a model performs on unseen data. A model that can only classify samples on which it has been trained on is not a very useful model. Furthermore, overfitting often means that the model is so large, having a high capacity, that it essentially is able to encode the information of the training data in the network [7], [14]. An analogy to this would be memorizing past exams. On the memorized exams performance would be great, but on unseen exams the performance would likely be lower since memorizing is not the same as learning.

There are several methods that can be used to reduce overfitting; two of the most popular ones are called ℓ_1 and ℓ_2 regularization [7]. ℓ_1 and ℓ_2 regularization also go under the name of *weight decay* [7]. Both methods add a penalty term to the loss function and the size of the penalty is decided via the alpha coefficient. ℓ_2 moves parameters close to zero, thus reducing the less important outputs to a further extent. Although ℓ_1 can also push the weights close to zero, it is also able to set some parameters to zero, completely excluding some outputs. Outputs in this context is in regards to the output of a single neuron in feedforward neural network. Therefore, penalties ℓ_1 and ℓ_2 can be seen as a way to prevent the model from memorizing the training data by reducing the capacity of the model as a result of limiting or removing outputs.

Early stopping is another technique that can be utilized during training of a model and act as regularization. Since training and validation performance is tracked, one can stop training the model if the performance on the validation set is either decreasing or stagnant and the performance on the training set is increasing or stagnant.

I. Loss Functions

The basis for learning with respect to ANNs is the loss function. Every parameter of the network is changed in regards to this loss via optimization with SGD and backpropagation. The loss function therefore represents the performance of the model since the optimization is to reduce the loss of the network. Depending on the task, loss functions vary, for tasks such as binary classification the cross-entropy loss is most common, seen in Equation (5). In Equation (5) \hat{y}_i is the

predicted value and y_i is the label. The label representing the correct value is either 1 or 0, while the predicted value can take on any value in the interval $[0, 1]$. Therefore, \hat{y}_i can be seen as the probability of class 1 and $1 - \hat{y}_i$ as the probability of class 0, from a binary classification perspective. Both \hat{y}_i and $1 - \hat{y}_i$ are in the range $[0, 1]$, resulting in the *log* of these values being ≤ 0 . If the label is of class 1, the second part of the expression is set to zero, and vice versa. N represents the number of samples.

$$L(\hat{\mathbf{y}}, \mathbf{y}) = -\frac{1}{N} \sum_{i=1}^N [y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)] \quad (5)$$

To get an idea of how the parameters of the model is changed in regards to this loss, \hat{y}_i should be dissected. \hat{y}_i is the output of the model and thus dependent on all parameters of the model, calculating the gradient is then the process of untangling the individual parameters effect on \hat{y}_i which in turn affect the loss.

Essentially cross-entropy, measures how far away the estimate is from the target for each class and then averages it to a final loss [18].

Taking into account the analogy presented in subsection II-G regarding solution spaces, the loss function can be seen as a sort of altimeter, a measure of how close to the bottom of the mountain range the explorer is.

J. Binary Classification

Binary classification [19] is a form of classification that only deals with two separate classes, often denoted as 0 (negative) or 1 (positive). The training data therefore consists of samples corresponding to either class 0 or class 1 and the task the model then needs to learn is to predict whether a sample belongs to class 0 or class 1. Examples of binary classification are diagnosis of a certain disease, since either the patient has a disease or not.

Since the task only deals with two different classes, there are four different types of predictions which can be made when compared to the actual label of a sample.

- 1) True positive(TP): Correct prediction of a class 1 sample.
- 2) True negative(TN): Correct prediction of a class 0 sample.
- 3) False positive(FP): False prediction of a class 1 sample.
- 4) False negative(FN): False prediction of a class 0 sample.

K. Binary Classification Metrics

When it comes to measuring the performance of a model trained on a binary classification task, there are many metrics that can be used [20]. The metrics that should be used depend on the goal.

Sensitivity, also known as true positive rate, is often used in combination with other metrics and represents how many of the positive samples that have been correctly classified.

$$TPR = \frac{TP}{TP + FN} \quad (6)$$

Precision (P or PPV) is another metric, representing the ratio of correctly classified class 1 samples to all predictions of class 1.

$$PPV = \frac{TP}{TP + FP} \quad (7)$$

F1 score is a metric representing the harmonic mean between precision and sensitivity. Therefore, this metric takes into account both aspects of sensitivity and precision, combining them into a single score.

$$F1 = 2 \cdot \frac{PPV \cdot TPR}{PPV + TPR} \quad (8)$$

All of the metrics mentioned above follow the principle that a higher score is the better score and the best possible score is one.

L. Transfer Learning

Transfer learning is an umbrella term for methods utilizing information from different contexts in order to improve model performance [21]. This often means that a model that has been trained to perform a certain task t_1 is used as a base in a different context on a different task t_2 . For example a model that has been trained to recognize dogs, represented as t_1 is later used to classify cats, represented as t_2 . The motivation behind this choice would be that the model that has learnt to classify dogs should have knowledge that is not only useful for classifying dogs. One could argue that the model has learned to recognize four legs and a tail, and since a cat also has four legs and a tail, this model can *transfer* knowledge from the first task to the second one [7].

Connected to this is the concept of pre-training. Given a task t_x where there is not a lot of available data, and a similar task t_y with lots of available data, pre-training can be leveraged. A model could first be trained to learn the task t_y and then utilize this knowledge when training on t_x [7].

When combining a model that has been trained with new layers which have not been trained, it is common practice to start of training only the new layers for a set amount of epochs. During these first epochs, the trained models parameters remain unchanged. This is because the new layers could disrupt what has been learned in the trained model, since large losses will occur at the beginning of training, essentially confusing the optimization algorithm. After the new layers have been trained, the whole model can be trained in unison, usually with a lower learning rate. This part of training is referred to as fine-tuning.

M. Ensemble Methods

Ensemble methods (or ensemble learning) is an umbrella term for methods utilizing multiple learning algorithms or models, often with the purpose of increasing performance compared to a single model [22]. There are many different methods, two of them being bootstrap aggregating (bagging) and stacking.

The motivation behind using ensemble methods can be compared to that it is often better to rely on more than one person or thing when making a decision.

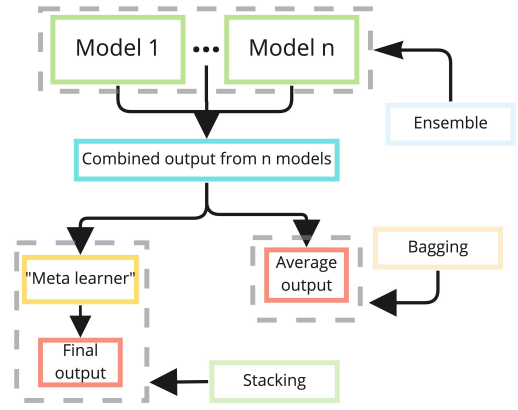


Fig. 2. Ensembles, bagging and stacking.

N. Bootstrap Aggregating (Bagging)

Bootstrap aggregating is a method that consists of two parts, first bootstrap and second aggregating. Bootstrapping is a statistical method of creating new subsets of a data set through sampling with replacement, meaning that there is a possibility of creating a new subset of data with some data points repeated. Bootstrapping is done to train models on slightly different versions of the original data set. This results in models with a higher probability of reaching a more diverse set of local minimums. Aggregating then consists of combining the outputs of the individual models and taking the average of this as the ensembles output [7]. This could be compared with asking a group of people to predict the number of balls in a glass jar, individual guesses are often far off and have a high variance when compared to each other, but as the number of people increases the average of the guesses reach a stable point. This is the goal of bagging. A visual representation can be seen in Figure 2.

O. Stacking

Stacking is an ensemble method which instead of taking the average of n number of models outputs relies on another model learning the optimal weights for combining these n models [14]. To avoid overfitting, this new "meta-learner" is trained on data which the base models have not been trained on. An example of stacking is to take the outputs of n ANN models trained on a binary classification task and feed the combined outputs into a logistic regression model. The stacked models prediction is then the output from the logistic regression, which has learned the optimal weights for combining the predictions of the n ANNs. Stacking is often used in competitions, as it almost always results in better performance than individual models, one example being the team taking second place in the Netflix Grand Prize as discussed in [23]. A visual representation of the stacking process can be seen in Figure 2.

P. Geometric Brownian Motion

Geometric Brownian Motion (GBM) is a stochastic process represented by Equation (9). GBM is often used in order to

simulate stock prices because of the similarities of the time series it produces [24].

$$S(t) = S(0) \cdot \exp\left(\left(\mu - \frac{1}{2} \cdot \sigma^2\right) \cdot t + \sigma \cdot B(t)\right) \quad (9)$$

$S(t)$ is the price, μ = drift, σ = volatility, and B is the Brownian motion. The Brownian motion, often referred to as the Wiener process, is a stochastic process with independent increments, meaning that future time steps are not dependent on previous ones. At each time step, the value of the process is changed by an observation x_i generated from a random variable X that has a normal distribution with a certain mean and variance.

Q. Scaling and Normalization

Feature scaling and normalization is an important part to consider when training learning algorithms. As the data can come in different sizes, it is important to scale them correctly so that they can be compared to one another. If scaling is not applied a feature with a high value can wrongfully impact the learning algorithm more than it should and vice versa. There are different types of scaling, but a common and straightforward one is the min-max normalization [25], as seen in (10). By taking the value and dividing it by the range and subtracting the minimum value of the range, you get a scaled value from $[0, 1]$. Depending on the problem formulation, this can be scaled to whatever range suits the problem.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (10)$$

R. Technical Indicators

Technical indicators represents a set of functions $I = \{f_1, f_2, f_3, \dots, f_n\}$ with the common factor between all function being that they extract new information from an existing time series, often based in statistics. The goal of a technical indicators is to provide some form of predictive power of the time series being analyzed. Technical indicators are most often used in fields related to finance [26], and the role they play depend on the context. In algorithmic trading the use of historical data for predictive purposes plays an important role in optimally executing trades [27], but there is indication of a greater adoption of technical indicators [28]. Nonetheless technical indicators are widely used by individual traders, hence their popularity.

An example of a technical indicator is the simple moving average (SMA), formulated in Equation (11). In this equation n denotes the number of previous time steps to consider, for example the number of days. x_i denotes the value of a variable on time step $k - i$, where k denotes the current time step.

$$SMA_n = \frac{1}{n} \sum_{i=1}^n x_i \quad (11)$$

SMA_5 can be said to belong to an infinite set of functions of the form SMA_n , essentially representing a family of functions. The infinite set comes from the fact that $n \in \mathbb{Z}^+$.

Technical indicators come in many different forms, most often dependent on some parameter, such as n in Equation

(11). This means that the set I is comprised of an infinite amount of functions, represented by n families of functions where $n \gg 0$.

III. PROBLEM FORMULATION

The problem that this thesis is trying to solve is of the form: Given a data set X containing information about n instances of a trade y_i and either implicitly or explicitly containing k instances of a non-trade z_i , is it possible to extract the underlying trading algorithm generating these trades y_i ? Furthermore, there is also the following problem: Given relatively few observations of the class y , what model and combinations of techniques give the best results?

Since there are only two classes in our data set, a trade y_i or a non-trade z_i , it is a binary classification problem.

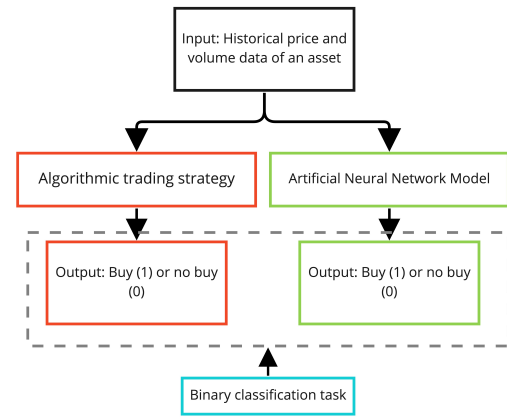


Fig. 3. Visualization of the problem formulation.

IV. METHOD

This section goes through the essential steps of how this thesis went about to solve the formulated problems. This includes information about data preparation and data set creation, model architecture, experiments and their designs, as well as motivation for choices that need further explanation. The general process of the project can be seen in the flowchart of Figure 4.

This project was to a large part done in Python with the help of the following Python libraries: TensorFlow and the Keras API implementation, NumPy and scikit-learn [29]–[32].

A. Data for Binary Classification

The data set which the models were trained, evaluated and tested on consisted of one strategy based solely on price and volume data. The exact nature of the trading strategy which generated the data set was not known by the authors during the projects lifetime, in order to limit bias in how the models were constructed and trained. The data set consisted of instrument name, date of trade and the time period for when signals were searched. From this information data sets were created for training, evaluating and testing of models.

The strategy that generated the data we were given was dependent on weekly price and volume data. The strategy

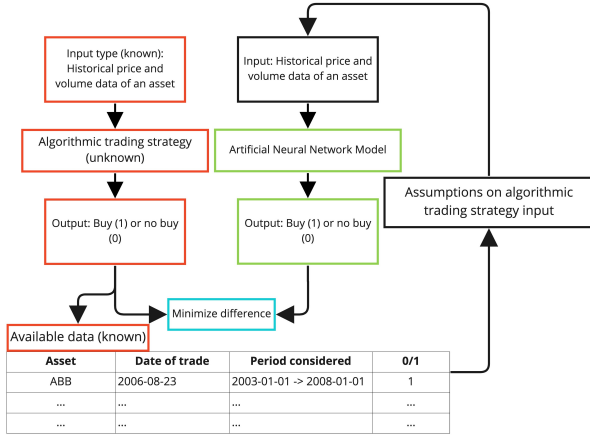


Fig. 4. Flowchart of the method process.

consists of four different criteria that all have to be met in order for a trade to be generated. The criteria are based on comparing a period of time spanning 2 years back from the week that is being evaluated. In total there are over 10 different statements comparing over 15 different values in differing ways, including data on open price, high price, low price, close price and volume all based on weekly data.

B. Feature Engineering

Often when building a machine learning pipeline, feature engineering plays an important role in improving model performance. This is because knowledge of domains relating to the task at hand can lead to creation of higher level features from the raw data, making the models job a bit easier.

The scope of this project, even after choosing to investigate only purely technical strategies, is quite large. The space of possible algorithms that could be employed to form a trading strategy based on technical indicators is infinite. Because of this, we chose not to employ feature engineering and instead focus on creating model architectures able to approximate wide function spaces together with other deep learning methods. Further motivations are that feature engineering is often time consuming and can lead to what is called *The curse of dimensionality* [7], essentially increasing the space of possible solutions to such a point that the data becomes scarce from a machine learning perspective. There are, however, many techniques to fight problems with too many features, such as ℓ_1 regularization for models and principal component analysis in a data preparation step.

The data which the models were trained on consisted therefore only of the following features: open price, high price, low price, close price and volume.

C. Transfer Learning and Simulated Data

This subsection is only relevant to the models which leverage transfer learning through a pre-training approach.

Transfer learning is a deep learning method which we decided to employ as an alternative to feature engineering, which we decided to avoid based on the reasons put forward in subsection IV-B.

The transfer learning approach consists in our case of pre-training large ANNs to predict n numbers of different time series. This stands as an alternative to feature engineering because of the nature of the time series that the models learn. The time series are generated by different technical indicators and families of these indicators. Therefore, the pre-trained models are trained to learn the functions that are generating these technical indicators, representing a subset of functions. The final model meant to perform the classification task of the original problem formulation is created by setting the pre-trained models on top of the new one. From this comes the motivation of utilizing pre-trained models instead of spending time on feature engineering, which can be summarized by these statements.

- 1) The algorithm which the binary classification task is trying to extract has a high probability of being dependent on some function(s) from the set of functions of technical indicators (since the strategies this project tries to recover are only based on historical price and volume data). They do not necessarily belong to the same subset of functions as the pre-trained model has learned or to the same families of functions, but the same set nevertheless.
- 2) Since functions in the same family are similar, often only distinguished by choice of parameter, the model should be able to learn these quickly. In addition to this, many families of functions are similar to each other, and therefore the model should be able to generalize faster to new families as well.

Since the pre-trained models require large amounts of data to learn these functions, we decided to train these models on simulated data instead of real data from stock exchanges. The simulated data was generated by utilizing Geometric Brownian Motion. The choice of parameters μ (drift) and σ (volatility) was made by estimating the drift parameter of 200 stocks from the Swedish stock exchange through simple statistical methods. This was done in order to generate time series similar to the price and volume data of real stocks. Since the simulated data does not necessarily need to emulate real data, the parameters were shifted from the estimations in order to create a time series on which the technical indicator functions produced results similar to that of real data.

D. Scaling and Sample Creation for Classification Task

In order to train ANNs, the data that the networks receive needs to be in a certain range for the underlying optimization algorithm to perform. This has to do with properties of gradient descent and calculation of derivatives together with backpropagation. There are many different methods that can be employed, depending on the type of data the model is to be fed. We chose to scale our inputs, across the entire time series, by the min-max method as seen in Equation (10). This is important because the strategies the models should be able to detect often rely on the relation between features, such as open price and low price. In this regard, the features related to the price data were scaled in unison to maintain their relation, and the volume data was scaled separately with the same method.

The end result is that each feature is transformed from the span $[x_i, y_i]$ (i denoting the min and max for each feature across a time series), to the span $[0, 1]$.

When it comes to creation of samples, we chose to create samples each containing 300 rows and 5 columns. The rows represent consecutive days and the columns each feature. Each sample has a single label that contains information about the last row, meaning the last time step of the time series, since we employ the many-to-one recurrent neural network procedure as described in the background. The models can therefore learn from the preceding 299 rows of data to classify the last step as a 1 or a 0.

E. Train, Validation and Test Set

When training models, some data need to be preserved to accurately determine performance during training and also to evaluate the model after training. Therefore, a training, validation, and test set needs to be established. We chose to set the training set of the data to contain 60% of the original data and the validation and test set each contained 20% of the original data. This was however adjusted when training the models on smaller subsets of the data. To get an accurate sense of model performance after training, the test set was much larger than the training set, as can be seen in the results section. See subsection IV-O.

F. Model Selection and Architecture

The models that were evaluated in this project consist of four different architectures, two of them leveraging different base models benefiting from transfer learning through pre-training. The first two models, labeled *naive LSTM* and *naive CNN-LSTM* represent an approach to model creation that is not the most sophisticated. Meaning that these models do not employ techniques such as transfer learning and their structure were not given much thought. These models act as a baseline for evaluating the two more sophisticated models, leveraging transfer learning and an architecture given more thought in context of the task at hand.

All of the models do however make use of being fed the raw features not only to the first layer but also to one of the final layers in order to preserve information that otherwise could have been lost. This comes from the assumption that most trading strategies involve comparing features, for example price of the current day to some technical indicator.

All of the models make use of the Adam optimizer.

G. Naive LSTM Model

The naive LSTM model consists of five LSTM layers with 50, 20, 20, 10 and 5 neurons in order from first to last. Every LSTM layer has the standard activation functions from the Keras API in TensorFlow. The model output layer is a dense layer consisting of a single neuron/unit with a sigmoid activation function, Equation (1) and binary cross-entropy as loss function, Equation (5).

H. Naive CNN-LSTM Model

The naive CNN-LSTM layer consists of four Conv1D layers followed by four LSTM layers and a dense layer just as the naive LSTM model. The Conv1D layers have 40, 30, 30 and 20 filters in order from the first to last layer and 10, 7, 5 and 2 as kernel size in the same order. The padding is set to casual and there is no activation function as this is the standard configuration for the Keras API TensorFlow implementation. The LSTM layers have in order from the first to last 20, 20, 10 and 5 neurons each. The dense layer and loss function is the same as for the naive LSTM model.

I. Small Pre-trained Model

The small pre-trained model consists of a base model with the following structure. The first layers make up a simplified version of the *WaveNet* [10] structure. The dilation of the layers is from first to last 1, 2, 4, 8, 16, 32, 64, 128 and 256. The filter for each layer is set to 30 and kernel size is set to 2, padding is set to causal and the activation function is set to ReLU. Following this is another Conv1D layer with number of filters set to 10 and kernel size of 1, no activation function. The next 3 layers, including the last one, are made up of LSTM layers with 15, 12 and 6 neurons. All these layers have the ReLU activation function. The number of neurons in the final layer represents the number of time series this base model is trained to predict.

The last part of the model consists of three LSTM layers with 50, 35 and finally 20 neurons from the first to last. The activation function of these layers are tanh, as seen in Equation (2). Following are two dense layers with 15 and 10 neurons, each with the ReLU activation, and finally an output layer with the same settings as the models above. The loss function is also binary cross-entropy.

J. Large Pre-trained Model

The large pre-trained model has the same starting structure as the small one, the difference being that the number of filters for each Conv1D layer is set to 80. Another difference is that the activation function for every layer except the last in the base model is Leaky ReLU; this was to combat the problem known as dying ReLU as mentioned in the background. After the Conv1D layers there are four LSTM layers each with 90 neurons and the tanh activation function. The last layer of the base model is a dense layer with 26 neurons and no activation function.

The last part of the model, after the pre-trained base model, has the exact same structure and settings as the last part of the small pre-trained model.

K. Binary Class Ratio Experiment

In order to decide the optimal ratio between the two classes, an experiment was constructed to find the optimal ratio of the classes as seen to model performance. Depending on the goal of the classification task and the distribution of the raw data the models are meant to classify, the ratio of samples corresponding to each class can be modified to construct the

training data. The best choice is often to have the training data represent the actual underlying distribution, but depending on the size of the data set and computational capacity, this might not always be the case.

This experiment was therefore set up in order to judge what ratio of classes gave the most optimal model performance on the test set. Due to time constraints, we chose to test the small pre-trained model for 40 epochs for each ratio, with no fine-tuning of the base model. The ratio of class 1, representing a trade, was 50%, 30%, 10%, 5% and 1%.

L. ℓ_1 and ℓ_2 Regularization Experiment

Since the pre-trained models are of significant size, an experiment was created in order to evaluate whether applying regularization to certain layers would improve the model performance. The reason being to prevent potential overfitting and to induce sparsity around the connection between the pre-trained base model and the final layers. Sparsity around these layers could benefit the models, since the base model has a high likelihood of providing irrelevant information for our problem. ℓ_1 regularization could thus act as a stopping mechanism, eliminating outputs from certain neurons and lead to better generalization. ℓ_2 regularization could in a similar fashion bring the outputs close to zero.

This experiment was conducted on the large pre-trained model, since that model would benefit the most from regularization. The regularized model was made up of a combination of ℓ_1 and ℓ_2 with a common coefficient value α . The model was then trained and evaluated on the following coefficient values α : 0, 0.0001, 0.001, 0.01, 0.05, 0.1, 0.2 and 0.4. The same training data were used for all model configurations with a ratio of 30% of class 1. The models were trained on 60 normal epochs and 60 fine-tuning epochs due to time constraints.

M. Data Set Size Experiment

In order to decide what models and techniques performed the best with respect to real-life applications, an experiment was conducted to evaluate the performance of the model on smaller subsets of the original data. The motivation behind this is that when faced with a problem of this kind in a real setting, data is likely not readily available and the data sets that can be constructed are likely small.

For this reason an experiment concerning data set size was constructed where all the models mentioned previously were evaluated, including a regularized version of the large pre-trained model (with alpha coefficient of 0.01). The different data sets contained between 3.8% to 5% of samples corresponding to class 1, a trade. The number of class 1 samples were 23, 49, 79 and 151, each representing a different subset of the original data set.

All models were trained with an upper limit of 800 epochs together with early stopping. Early stopping had a 30 epoch window and the best weights as seen from the validation data set were kept. The pre-trained models were then fine-tuned with an upper limit of 800 additional epochs with the same early stopping method. This ensured that all the models had a chance to reach their optimum performance.

N. Ensemble Methods Experiment

As the goal of this project was to determine which models and which techniques give rise to the best performance, we set up an experiment to evaluate whether ensemble method techniques could be relevant.

This experiment consisted of training five and ten different models on different subsets of the same underlying data set, also called bagging. Each subset had around 5% of class 1 samples and the original data set had 151 class 1 samples. The individual model performance could then be compared to the created ensemble of models.

The experiment also included an evaluation of an ensemble method technique called stacking. For this an additional model, which can be regarded as a "meta-learner", was trained on additional data. The meta-learner model was trained on 25, 50, 75 and 150 additional class 1 samples with a 5% ratio of class 1 samples. The meta-learner model was a logistic regression model from the scikit-learn library, with default parameter settings.

O. Test Set

The test set on which all the models were evaluated has the following characteristics. The test set contains 89666 samples, each sample corresponds to a data set of 300 rows with 5 columns as described in subsection IV-D. Each sample has a label that is class 1, representing a trade, or class 0, representing no trade. Of the 89666 samples, 269 of them have class 1 labels and 89397 samples therefore have class 0 labels. The choice of constructing a test set with these characteristics comes from evaluating the underlying data set which we were given. We assumed that the strategy generating these samples was based on daily data, and from this assumption we created the test set. The assumption of daily data meant that we interpreted every day of the interval specified in the underlying data as a potential trade and from this we calculated the number of potential trades. From this a ratio could be calculated that describes the frequency of actual trades. This ratio turned out to be 0.3%, meaning that 99.7% of all possible days did not produce any trade.

V. RESULTS

The following subsections will contain the results of each experiment introduced in subsections IV-K to IV-N. The evaluation of every model for all experiments will be on the test set as described in subsection IV-O. *nan* denotes missing values, the reason often being division by zero. In Tables I-VI the column P denotes the metric *precision*, that is the percentage of true predicted trades of all predicted trades.

The models will in the tables below have abbreviations. The naive LSTM model will go by L, naive CNN-LSTM will go by CL, small pre-trained as SP, large pre-trained as LP and regularized large pre-trained with alpha coefficient of 0.01 will go by LPR.

Other abbreviations for metrics and such can be seen in subsection II-J and II-K.

A. Binary Class Ratio Experiment

The column ratio denotes the percentage of class 1 (trades), contained in the training set of the corresponding model. These results can be seen in Table I.

TABLE I
VARYING CLASS RATIO EVALUATION OF MODEL SP

Model	ratio (%)	TP	FP	TN	FN	P	F1
SP	49.5	224	11973	77424	45	0.018	0.036
SP	30.2	227	11772	77625	42	0.019	0.037
SP	10.6	142	4470	84927	155	0.031	0.058
SP	5.1	114	2087	87310	155	0.052	0.092
SP	1	0	0	89397	269	nan	0

B. ℓ_1 and ℓ_2 Regularization Experiment

Information about the training and training data can be found in subsection IV-L. The results of this experiment are displayed in Table II.

TABLE II
REGULARIZATION EVALUATION ON LP

Model	alpha	TP	FP	TN	FN	P	F1
LP	0	221	12454	76943	48	0.017	0.034
LP	0.0001	226	11118	78279	43	0.020	0.039
LP	0.001	221	10290	79107	48	0.021	0.041
LP	0.01	231	9795	79602	38	0.023	0.045
LP	0.05	227	9807	79590	42	0.023	0.044
LP	0.1	220	9139	80258	49	0.024	0.046
LP	0.2	224	9819	79578	45	0.022	0.043
LP	0.4	0	0	89397	269	nan	0

C. Data Set Size Experiment

in Table III, n_1 denotes the number of class 1 instances, trades, contained in the data set which the corresponding model was trained on. n_0 denotes in the same fashion the number of class 0 instances, non-trades.

D. Ensemble Methods Experiment

in Table IV n_1 and n_0 denotes the same information as in Table III. Each data set, represented by n_1 and n_0 , was created through bootstrapping of the original data set with the following characteristic $n_1/n_2 = 151/2880$. The model labeled E_5 is the average of the first five models as seen in Table IV and E_{10} is the average of *all* individual models in Table IV, following the bagging method.

The following tables, V and VI, contain results of applying the stacking method on the individual models seen in Table IV. Table V evaluates stacking on the first 5 models and Table VI evaluates stacking on all 10 individual models. n_{extra} denotes the extra number of class 1 samples, trades, on which the meta-learner model was trained as explained in subsection II-O. SM is the name of the meta-learner model.

TABLE III
DATA SET SIZE EVALUATION ON ALL MODELS

Model	n_1/n_0	TP	FP	TN	FN	P	F1
LP	23/583	0	0	89397	269	nan	0
LP	49/1164	0	0	89397	269	nan	0
LP	79/1740	0	0	89397	269	nan	0
LP	151/2880	31	396	89001	238	0.073	0.089
LPR	23/583	19	1904	87493	250	0.010	0.017
LPR	49/1164	24	1099	88298	245	0.021	0.035
LPR	79/1740	38	1070	88327	231	0.034	0.055
LPR	151/2880	39	761	88636	230	0.049	0.073
SP	23/583	0	0	89397	269	nan	0
SP	49/1164	0	0	89397	269	nan	0
SP	79/1740	7	232	89165	262	0.029	0.028
SP	151/2880	26	377	89020	243	0.065	0.077
L	23/583	0	0	89397	269	nan	0
L	49/1164	0	0	89397	269	nan	0
L	79/1740	0	0	89397	269	nan	0
L	151/2880	0	0	89397	269	nan	0
CL	23/583	0	0	89397	269	nan	0
CL	49/1164	0	0	89397	269	nan	0
CL	79/1740	0	0	89397	269	nan	0
CL	151/2880	0	0	89397	269	nan	0

TABLE IV
BAGGING EVALUATION

Model	n_1/n_0	TP	FP	TN	FN	P	F1
LP ₁	157/2874	9	130	89267	260	0.065	0.044
LP ₂	153/2878	29	611	88786	240	0.045	0.064
LP ₃	174/2857	56	1131	88266	213	0.047	0.077
LP ₄	155/2876	0	0	89397	269	nan	0
LP ₅	149/2882	30	918	88479	239	0.032	0.049
E_5	nan	19	218	89179	250	0.080	0.075
LP ₆	161/2870	48	1913	87484	221	0.025	0.043
LP ₇	142/2889	22	665	88732	247	0.032	0.046
LP ₈	158/2873	9	104	89293	260	0.080	0.047
LP ₉	153/2878	5	130	89267	264	0.037	0.025
LP ₁₀	143/2888	33	687	88710	236	0.046	0.067
E_{10}	nan	13	197	89200	256	0.062	0.054

TABLE V
STACKING EVALUATION 5 MODELS

Model	n_{extra}	TP	FP	TN	FN	P	F1
SM	25	13	161	89236	256	0.075	0.059
SM	50	25	355	89042	244	0.066	0.077
SM	75	24	337	89060	245	0.067	0.076
SM	150	38	618	88779	231	0.058	0.082

TABLE VI
STACKING EVALUATION 10 MODELS

Model	n_{extra}	TP	FP	TN	FN	P	F1
SM	25	29	444	88953	240	0.0613	0.078
SM	50	32	447	88950	237	0.0668	0.086
SM	75	29	397	89000	240	0.0681	0.084
SM	150	42	697	88700	227	0.0568	0.083

VI. DISCUSSION

In order to discuss the evaluation result of each experiment that was conducted, a baseline for model performance should

be set. If the models actually have learned something, their predictions should not be random. Since the test set that all models were evaluated on contains 0.3% class 1 instances, predictions of class 1 would be random if the following fraction is less than 0.3%: $\frac{TP}{TP+FP}$. This comes from the fact that if the predictions were truly random, there is a 0.3% of the predictions corresponding to an actual class 1 sample. As can be seen in subsection II-K, this fraction corresponds to the binary classification metric precision, included in all tables of results of the experiment as P. If the models do not reach above this threshold, they should not be considered any further. However, if they reach above this threshold, further discussions of their performance are relevant.

The aim of this project was to evaluate whether the models are capable of extracting the underlying trading algorithm. In this regard the precision metric is lacking, since the number of correctly predicted actual trades is of high importance. For this reason we choose to evaluate our models on the F1 metric.

A. Binary Class Ratio Experiment

The results from this experiment can be seen in Table I. As the training set converges to the ratio of the test set both precision and F1 increases. However, when the training set only consists of 1% of class 1 samples, it does not classify any sample as a trade. There could be several reasons for this. First of all, as the ratio of class 1 samples decreases the number of class 0 samples increases together with the data set itself, leading to the model being exposed to fewer and fewer class 1 samples per batch (stochastic gradient descent). Since the models were only trained on 40 epochs, it could be that the model would start classifying class 1 samples after another n epochs, since it would be exposed to more class 1 samples. Another reason could be that when the training data set gets more skewed toward class 0 samples, the need for bigger data sets increases in order for the model to get an accurate understanding of the underlying process. Additionally, improvements could be seen if the batch size was increased in order to incorporate more examples of class 1 during each batch.

B. ℓ_1 and ℓ_2 Regularization Experiment

The results from this experiment can be seen in Table II. The results indicate that as regularization increases, the performance increases in unison up to a certain point. Even if we did not train multiple models for every value of alpha, there is indication of a plateau in performance at a value of 0.01 and increasing. The results show that incorporating regularization, at least in large models, can lead to an increase in performance. It should, however, be stated that since the models were only trained for a total of 120 epochs, performance of the models could change if let to train further. Furthermore, the performance difference between the models is not large enough to determine an optimal value for alpha.

C. Data Set Size Experiment

The results of this experiment can be seen in Table III. From the data set size experiment we see that the two naive

models are not able to identify any class 1 samples on all data sets that the models were evaluated on. It should be said, that even though these models did not manage to determine any trades on the smaller subsets, they did on the bigger sets. These results were not included in the report. An interesting observation is that the regularized large model was the only one able to classify class 1 samples on the smallest subsets.

The large pre-trained model performed best in terms of both precision and F1 score in the largest data set of the experiment, outperforming the regularized version by a noticeable margin. The smaller pre-trained model also outperformed the regularized large model, but only on the biggest data set and not to the same extent. These observations could point to the regularized models being a good choice when dealing with small data sets. A reason for this could be, as explained in subsection II-H, that the regularized models are able to quickly eliminate connections or push the most unnecessary connections close to zero. Leading to extraction of the most relevant information sooner. There is, however, only so much information in small data sets. The large pre-trained model reaching a precision of 7.26% on a small data set together with the fact that ANNs are usually trained on large amounts of data in order to perform. These results are promising and indicate that models of this type are able to learn complex algorithms to a certain extent, even if data is scarce.

D. Ensemble Methods Experiment

The results from final experiment of this project can be seen in Tables IV, V and VI.

The results of the bagging experiment in Table IV do not provide a clear enough result to decide whether bagging actually improves performance compared to individual models. However, the ensemble model consisting of the first five models outperformed all the individual models in regards to precision and achieves an F1 score slightly below the best individual model. The ensemble model consisting of all of the individual models does underperform several of the individual models, both in regards to precision and F1 score.

There are benefits which precision and the F1 metric do not show, and those relate to variance in the predictions. As the number of individual models increases, the ensemble model converges to become a sort of mean of all possible models, combining information from a wide range of local minimum's in the space of possible solutions. The goal of the project was not to create a model with these characteristics, and therefore this approach to bagging can not be said to increase performance. Apart from bagging, there are other interesting ensemble techniques which do not only take the average of the predictions but instead rely on different model criteria. An interesting example would be to combine individual models based on their performance with regard to certain metrics. For example, combine the ones with highest precision and the ones having the highest sensitivity. A fitting analogy would be a sort of selective "breeding" process of models, with some specific goal in mind.

The results from the stacking experiments indicates that stacking is good choice if the goal is to increase the F1 score.

With 5 models, as displayed in Table V, the stacked ensemble performs at first worse than the ensemble with 5 models leveraging only bagging. At 50 extra class 1 samples the F1 score does reach above the mentioned model, but precision does decrease. When using 10 models, the results shown in Table VI, the improvements compared to only using bagging are clear even at only 25 extra class 1 samples. The F1 score is 32% higher than the bagging model consisting of 10 models and higher than any individual model. At 50 extra samples the F1 score increases an additional 10% as compared to with 25 extra class 1 samples, and precision increases as well. The trend does not, however, seem to continue as the number of extra samples increases from this point. This could be due to the fact that there are only a few weights and biases to be learned. Stacking not requiring large amounts of new data is important since available data is limited. Aside from the positive results, there is the question of whether the individual models would perform even better than the stacked model, if they were trained on these extra samples instead. This factor should be tested to conclude that stacking is actually a method that should be used. But there is also the possibility that stacking would provide better results even if the meta-learner is not fed completely new data. For example, the validation set from the neural network training could act as the training set for the logistic regression model. This would make stacking an attractive option when it comes to problems such as this.

E. Model Architecture Choice

Regarding the choice of model architecture, it is not completely clear whether the approach to transfer learning actually benefited the models the way it was intended. There are, however, clear indications that the larger models perform better than the smaller ones. In order to properly evaluate what model architecture is best suited for the problem, hyper-parameter optimization should be done. The search for hyper parameters could then include different base models, some being pre-trained, and others not. This process is time-consuming, especially with large models, which is why it was not used in this project. Further evaluations regarding pre-training of models should also include multiple different trading algorithms, since the motivation behind using transfer learning was creating models capable of adapting to a wide range of potential strategies.

F. Trading Algorithm and Data

The trading algorithm which the models were meant to learn was not known during the project lifetime to reduce bias. As can be seen in subsection IV-A, the algorithm that produced the samples is advanced. This trading algorithm was furthermore based on weekly data and utilized data going as far back as 2 years from the week being evaluated. The models were trained on data containing 300 rows, each row representing a day, meaning that not only were the models fed daily data instead of weekly but also lacked an entire year of relevant information. With this in mind, the performance of the models, especially in the smaller subsets, was impressive. The fact that the algorithm was based on weekly data most likely

means that more trade samples would have been generated if the algorithm had instead been based on daily data. This is because the algorithm would evaluate more points of data and probably produce more trade instances, from a probabilistic perspective. Therefore, this could, in some sense, mean that some of the false positives the models predicted could be actual trades seen from a daily perspective.

Evaluation of the models on different time spans, such as weekly or monthly, would not have been a challenging task. It is likely that the model performance would have improved if trained on weekly data instead of daily. It would also have been simple to train the models on bigger samples, which could have led to a boost in performance. These are factors that should be taken into account in future work regarding the theme of this thesis.

G. Applications and Future Work

The scope of this project was reduced to solving a problem that can be regarded as feasible in some sense. Because of this, the models that were created are not likely able to generalize to a large amount of possible trading algorithms. This is because trading algorithms can utilize a wide variety of data, as mentioned in the introduction. There is still something to be said about creating models capable of learning relatively complex strategies, even if the data are constrained to only historical price and volume data. Future research in this area could take a similar approach as this thesis, since there is a high chance of trading algorithms sharing commonalities, even if they rely on different types of data. One could, for example, include a feature containing macro economic information or/and sentiment analysis extracted from, for example articles. Thus, this thesis could be seen as a general evaluation of the models capabilities to extract algorithmic features.

As mentioned previously, future work could include a thorough hyper-parameter search for optimal model architectures and optimal creation of data sets. This would be a time-consuming task and should be done, for example, by using distributed training, cloud computing, and other methods aimed at speeding up neural network training. Future work on this topic could also use the results of this thesis to limit the scope of a problem. The results in regards to larger pre-trained models on small data sets, the use of regularization and stacking give strong indications of improved performance and could thus act as a potential baseline for further studies.

VII. CONCLUSION

To conclude, the results show that some of the models are able to mimic the underlying trading algorithm to some extent, even if it is complex and fed data from both a longer period and of a slightly different sort from the true data. Many models performed much better than purely random guesses of a trade, even when trained on small data sets.

The larger pre-trained models achieved the highest performance, indicating that transfer learning should be used to increase performance.

The ensemble method stacking elevates performance compared to that of the individual models when trained on a

small amount of extra data. Including regularization in larger models hint towards being an optimal choice when dealing with very small data sets, but the inclusion of regularization when dealing with larger data sets is not clear.

ACKNOWLEDGMENT

The authors thank their supervisor Javad Parsa for his support in the project and general good mood!

REFERENCES

- [1] E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *The Journal of Finance*, vol. 25, no. 2, pp. 383–417, May 1970. [Online]. Available: <http://www.jstor.org/stable/2325486>
- [2] A. P. Chaboud, B. Chiquoine, E. Hjalmarsson, and C. Vega, "Rise of the machines: Algorithmic trading in the foreign exchange market," *The Journal of Finance*, vol. 69, no. 5, pp. 2045–2084, Oct 2014. [Online]. Available: <http://www.jstor.org/stable/43612951>
- [3] G. Zuckerman. (2020, April) Renaissance's \$10 billion medallion fund gains 24% year to date in tumultuous market. [Online]. Available: <https://www.wsj.com/articles/renaissance-s-10-billion-medallion-fund-gains-24-year-to-date-in-tumultuous-market-11587152401>
- [4] H. Markowitz, "Portfolio selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, Apr. 1952. [Online]. Available: <http://www.jstor.org/stable/2975974>
- [5] R. Wigglesworth. (2018, April) How a volatility virus infected wall street. [Online]. Available: <https://www.ft.com/content/be68aac6-3d13-11e8-b9f9-de94fa33a81e>
- [6] A. Persson and R. Li, "Inversion of markowitz portfolio optimization to evaluate risk," BSc. thesis, KTH, Stockholm, Sweden, 2021.
- [7] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MA: MIT Press, 2016, <http://www.deeplearningbook.org>.
- [8] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterton, Eds., vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 249–256. [Online]. Available: <https://proceedings.mlr.press/v9/glorot10a.html>
- [9] 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.
- [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*, Sep. 2016. [Online]. Available: <https://arxiv.org/abs/1609.03499>
- [11] A. Gron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 1st ed. O'Reilly Media, Inc., 2017.
- [12] R. C. Staudemeyer and E. R. Morris, "Understanding lstm – a tutorial into long short-term memory recurrent neural networks," *arXiv*, Sep. 2019. [Online]. Available: <https://arxiv.org/abs/1909.09586>
- [13] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, Mar. 2013, pp. 6645–6649.
- [14] K. P. Murphy, *Machine learning : a probabilistic perspective*, Cambridge, MA, 2012.
- [15] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," *arXiv*, vol. abs/1811.03378, Nov 2018. [Online]. Available: <https://arxiv.org/abs/1811.03378>
- [16] L. Lu, "Dying ReLU and initialization: Theory and numerical examples," *Communications in Computational Physics*, vol. 28, no. 5, pp. 1671–1706, Jun 2020. [Online]. Available: <https://doi.org/10.4208%2Fci.2020-0165>
- [17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," vol. abs/1412.6980, Dec 2014. [Online]. Available: <https://arxiv.org/abs/1412.6980>
- [18] K. Janocha and W. M. Czarnecki, "On loss functions for deep neural networks in classification," *arXiv*, Feb 2017. [Online]. Available: <https://arxiv.org/abs/1702.05659>
- [19] R. Kumari and S. Srivastava, "Machine learning: A review on binary classification," *International Journal of Computer Applications*, vol. 160, pp. 11–15, Feb 2017.
- [20] G. Canbek, S. Sagioglu, T. T. Temizel, and N. Baykal, "Binary classification performance measures/metrics: A comprehensive visualized roadmap to gain new insights," pp. 821–826, Nov 2017.
- [21] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. PP, pp. 1–34, Jul 2020.
- [22] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *Journal of Artificial Intelligence Research*, vol. 11, pp. 169–198, aug 1999. [Online]. Available: <https://arxiv.org/abs/1106.0257>
- [23] J. Sill, G. Takacs, L. Mackey, and D. Lin, "Feature-weighted linear stacking," *arXiv*, Nov 2009. [Online]. Available: <https://arxiv.org/abs/0911.0460>
- [24] K. Reddy and V. Clinton, "Simulating stock prices using geometric brownian motion: Evidence from australian companies," *The Australasian Accounting Business and Finance Journal*, vol. 10, pp. 23–47, Sep. 2016.
- [25] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance," *Applied Soft Computing*, vol. 97, p. 105524, Dec. 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1568494619302947>
- [26] C. J. Neely, D. E. Rapach, J. Tu, and G. Zhou, "Forecasting the equity risk premium: The role of technical indicators," *Management Science*, vol. 60, no. 7, pp. 1772–1791, Jul. 2014. [Online]. Available: <http://www.jstor.org/stable/42919633>
- [27] J. Lorenz and R. Almgren, "Mean–variance optimal adaptive execution," *Applied Mathematical Finance*, vol. 18, no. 5, pp. 395 – 422, Oct. 2011. [Online]. Available: <https://search.ebscohost-com.focus.lib.kth.se/login.aspx?direct=true&db=bsh&AN=67129587&site=ehost-live>
- [28] T. Salkar, A. Shinde, N. Tamhankar, and N. Bhagat, "Algorithmic trading using technical indicators," in *2021 International Conference on Communication information and Computing Technology (ICCICT)*, Jun. 2021, pp. 1–6.
- [29] 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/>
- [30] F. Chollet *et al.* (2015) Keras. [Online]. Available: <https://keras.io>
- [31] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. [Online]. Available: <https://doi.org/10.1038/s41586-020-2649-2>
- [32] 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.

Estimating Believed Knowledge of Portfolio Agents Using Inverse Optimization

Gustaf Zachrisson and Oscar Wink

Abstract—In this report, we demonstrate the utility of inverse optimization in convex programming by applying it on estimating financial market beliefs and behaviors of portfolio investors. The inversion of the optimization utilized the Karush–Kuhn–Tucker optimality conditions specified for the current situation. The investor situation was simulated using the Markowitz model for optimal portfolio selection. Three model-specific implementations of inverse optimization were evaluated and the estimates were assessed by applying them in a solution to a portfolio agent sorting problem. The solution was perturbed with noise to test the robustness of the model. The work concludes that estimation by inverse optimization of Markowitz models is possible to a satisfactory degree but requires case-specific model design.

Sammanfattning—I den här rapporten demonstreras användbarheten av inverterad optimering för konvexa problem genom att applicera det vid skattning av investerares beteenden och förväntningar av en finansiell marknad. Inverteringen av optimeringsproblemet gjordes med hjälp av Karush–Kuhn–Tucker-villkor specificerade för det aktuella fallet. Markowitz-modellen användes för att modellera en finansiell marknad och val av optimal portfölj av investeringar. Tre modell-specifika versioner av inverterad optimering tillämpades och deras skattningar utvärderades genom att applicera dem i lösning av ett problem där portföljägare skulle sorteras. Lösningssättet exponerades för brus för att testa modellens robusthet. Slutsatsen som görs är att tillfredsställande skattning med inverterad optimering av parametrar i Markowitz-modellen är möjligt, men kräver ändamålspecifik design av modellen.

Index Terms—Inverse optimization, Convex Programming, Estimation, Operations Research, Markowitz Model, Portfolio Optimization

Supervisors: Jacob Lindbäck

TRITA number: TRITA-EECS-EX-2022:134

I. INTRODUCTION

A. Background

Who wouldn't want to make an optimal decision? Yet sometimes, it might come at the cost of exposing the decision-maker. A variety of situations have multiple possible choices with constraining trade-offs or consequences, of which an optimal choice is desired. Today, countless decisions are made continuously: not only by humans and animals, but by computing machines which our society has become heavily integrated with. The intrinsic strive for efficiency and comfort has always caused choices to emerge. With the future predicted to be even more digitalized, the number of decisions will increase for further streamlining.

However, if an optimal choice is motivated by an analysis of the situation, the choice itself could contain information about the situation – and potentially about the decision-maker. The

information available for an agent when making an optimal decision could then be retrieved by looking at the decision made.

This idea can be realized by what is known as inverse optimization, where the parameters characterizing the original system are estimated. Because of its generality, it is applied in diverse mathematical fields such as operations research, statistical inference or machine learning. With some knowledge and appropriate assumptions of how to model a situation robust enough, inverse optimization becomes applicable for sub-optimal situations and thus relevant for many real-world situations.

In this work, inverse optimization and parameter estimation are applied to financial asset portfolios. Portfolio selection is a grateful situation for optimization applications due to the existence of both proper mathematical models and abundance of available data.

Investors in financial markets wish to maximize return while balancing the risk of their investments. In general a low-risk investment will likely result in a profit, but a small one. An example of low-risk investments are assets in government or corporate bonds, yielding slightly more than a regular savings account. On the contrary, high-risk investments would more likely result in a loss of capital but the potential return is much higher. Examples of high-risk investments are crypto assets and assets in hedge funds.

In 1952 the American economist Harry Markowitz published a theory for asset portfolio allocation today called the Markowitz model [1]. The Markowitz model is based on an agent's beliefs about the financial market and how risk-averse the agent is. Markowitz was in 1990 awarded the Nobel Memorial Prize in Economic Sciences for the invention of the model, and it is today an integral part of the broader econometric framework called Modern Portfolio Theory.

While many investors have similar beliefs about the market, their risk aversion can vary and thus result in differently allocated portfolios. Furthermore, although many investors share a common belief this may not be true for all agents. For these situations, it might be of interest to identify the agents having significantly different market beliefs.

Previous research has been done in the area of inverse optimization, such as [2], [3] or reviewed in [4]. Likewise, the intersection of inverse optimization applied in portfolio selection has previously been examined such as in [5], [6] and [7]. Moreover, [7] elaborates on parameter estimation as a tool to investigate an agent's private believed knowledge instead of acquired objective knowledge. This addresses aspects of personal integrity in possible applications of inverse optimization.

B. Aim of Work

The model implemented and evaluated in this work is assigned multiple known optimal Markowitz portfolios. The aim for the model is to identify and sort portfolio owners based on estimations of their respective believed market state and individual risk aversion. These two parameters were chosen to be estimated due to them being of most interest and to limit the scope of the work. The model is also desired to be applicable for problems relevant for portfolio investments. Hence the work includes testing of correctness and robustness of the model.

C. Report Outline

Subsections II-B through II-E introduces theoretical mathematical concepts relevant for interpreting the results and understanding the method implemented to acquire them. The Markowitz model is introduced in detail in Subsection II-F and its mathematical properties in Subsection II-G.

The implementation of the parameter estimation is described in Section III along with the experiments designed to evaluate the correctness and robustness of the estimation. Section IV presents the outcomes of the conducted experiments. Section V includes interpretation and comparison of the outcomes with the underlying theory as well as analysis of the used mathematical models and their implementation. Finally, Section VI presents the main takeaways of the work.

II. PRELIMINARIES

A. Notation

In this report, definitions of quantities and their notation will be presented as they are introduced. The sections are arranged so that referred variables and equations have been introduced in previous sections. Below are some miscellaneous notations presented.

Throughout this report, the characters i and k will be used to denote indexes or iterations in varying situations. In general, lower-case characters such as j , n , m or p will be used consistently to denote a value, while Greek letters such as μ , γ , Σ or θ represents parameters or quantities.

Functions will be written with accompanying parentheses, where the optimal value for their argument variables will be notated with an asterisk, $*$.

Since some of the theories applied in this work are conditioned by their mathematical sets, two relevant sets are introduced below as defined in [8].

- 1) Affine sets $\mathcal{A} \subseteq \mathbb{R}^n$ can be defined as a set containing every affine combination of its points: any set of points $(x_1, \dots, x_k) \in \mathcal{A}$ can be linearly combined as an affine combination $x_0 = \theta_1 x_1 + \dots + \theta_k x_k$ where $\theta_1 + \dots + \theta_k = 1$. Then $x_0 \in \mathcal{A}$.
- 2) Convex sets $\mathcal{C} \subseteq \mathbb{R}^n$ can be defined as a set containing every convex combination of its points: any point (x_1, \dots, x_k) can be linearly combined as an convex combination $x_0 = \theta_1 x_1 + \dots + \theta_k x_k$ where $\theta_1 + \dots + \theta_k = 1$ and $\theta_i \geq 0$, $i = 1, \dots, k$. Then $x_0 \in \mathcal{C}$. For example, every affine set is also convex.

B. Optimization

A general optimization problem can be written on the form

$$\begin{aligned} &\text{minimize} && f_0(x) \\ &\text{subject to} && g_i(x) \leq 0, \quad i = 1, \dots, m \\ &&& h_i(x) = 0, \quad i = 1, \dots, p \end{aligned} \quad (1)$$

where $f_0(x)$, $g_i(x)$ and $h_i(x)$ are $\mathbb{R}^n \rightarrow \mathbb{R}$ functions. The objective function $f_0(x)$ is to be minimized depending on the argument $x = (x_1, \dots, x_n)$, acting as the decision variable. The set of possible (called feasible) decisions x are limited by constraint functions $g_i(x)$, $h_i(x)$ into the feasible set

$$\mathcal{F} = \{x \in \mathbb{R}^n \mid g_i(x) \leq 0 \forall i \in \mathcal{I}, h_i(x) = 0 \forall i \in \mathcal{E}\},$$

where \mathcal{I} and \mathcal{E} denotes the sets of inequality and equality constraints respectively [8]. The optimal decisions are then denoted as

$$x^* \in \arg \min_{x \in \mathcal{F}} f_0(x). \quad (2)$$

Condensed notation of (1) then becomes

$$\min_x f_0(x), \quad \text{s.t. } x \in \mathcal{F}. \quad (3)$$

C. Convex Optimization

Convex functions satisfies the inequality

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y), \quad (4)$$

with $x, y \in \mathbb{R}^n$, $\alpha, \beta \in \mathbb{R}$ and $\alpha, \beta \geq 0$ where $\alpha + \beta = 1$. Hence convex problems are more general than, and includes, linear problems [8], which analogously can be written as

$$f_i(\alpha x + \beta y) = \alpha f_i(x) + \beta f_i(y). \quad (5)$$

Convex optimization problems are a sub-type of nonlinear optimization problems where both the objective function and the feasible set are convex. Thus, the feasible set $\mathcal{F} \subseteq \mathbb{R}^n$ is a convex set and $f_0 : \mathcal{F} \rightarrow \mathbb{R}$ is a convex function.

For functions which are two times differentiable, convexity can be easily proved by examining the Hessian: the Hessian of convex functions is positive semi-definite and for strictly convex positive definite.

Furthermore, for convex optimization problems [9] states that

- 1) If there are multiple optimal choices, then the set of optimal choices $x^* \in \mathcal{F}^{opt}$ is convex and has infinitely many elements.
- 2) If f_0 is strictly convex, there exist at most one optimal choice.
- 3) Any local minimum is also a global minimum.

D. Optimality Conditions

For any optimization problem, conditions can be stated for its optimal decisions. For non-linear optimization problems, the so called Karush–Kuhn–Tucker (KKT) conditions need to be satisfied by necessity for any optimal decision under some simple regularity conditions.

The KKT conditions are five conditions which understanding can be facilitated by inspecting the Lagrangian function:

The Lagrangian function, $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$, of an optimization problem can be written as

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^m \lambda_i g_i(x) + \sum_{i=1}^p \nu_i h_i(x), \quad (6)$$

where λ_i and ν_i are Lagrange multipliers weighting the corresponding constraints $g_i(x)$ and $h_i(x)$ in (1).

Now, the following five KKT-conditions in (7-9) can be stated for any x to be an optimal decision to $\min_x f_0(x)$.

The optimal choice x^* also minimizes the Lagrangian function in (6) at optimal (λ^*, ν^*) . This is a stationary point where $\nabla_x L(x^*, \lambda^*, \nu^*) = 0$ and thus (7) must hold. This is called the stationary condition.

$$\nabla_x f_0(x^*) + \sum_{i=1}^m \lambda_i^* \nabla_x g_i(x^*) + \sum_{i=1}^p \nu_i^* \nabla_x h_i(x^*) = 0 \quad (7)$$

Furthermore, x need to satisfy primal feasibility conditions ($x \in \mathcal{F}$) and thus (8) must hold.

$$\begin{aligned} g_i(x^*) &\leq 0, \quad i = 1, \dots, m \\ h_i(x^*) &= 0, \quad i = 1, \dots, p \end{aligned} \quad (8)$$

Likewise, λ^* need to satisfy the dual feasibility condition but also the required complementary slackness that $\lambda^* g_i(x^*) = 0$ if $L(x^*, \lambda^*, \nu^*)$ is to be minimal. Thus (9) must hold.

$$\begin{aligned} \lambda_i^* &\geq 0 \quad i = 1, \dots, m \\ \lambda_i^* g_i(x^*) &= 0 \quad i = 1, \dots, m \end{aligned} \quad (9)$$

For a convex optimization problem – requiring a convex objective function, affine inequality constraints and convex equality constraints – the KKT conditions are sufficient for an optimal decision. In addition, if strict feasibility holds, the KKT conditions are also necessary. [8]

E. Inverse Optimization

For any model, its optimal decisions and its data are coupled. Viewing the objective function and the constraints as expressions depending on multiple parameters, one can choose which parameter to set as decision variable and which parameter to set as constants.

Once an optimization problem is solved, its optimal choice can be viewed as a known constant. Now a previously constant parameter can be set as the unknown decision variable for an inverse optimization problem: retrieving the model data based on a known optimal choice.

The original optimization problem of a model, called the Forward Optimization Problem model, can be denoted as

$$FOP(\theta) := \min_{x \in \mathcal{F}(\theta)} f(x, \theta), \quad (10)$$

where $\theta \in \Theta$ is a set of parameters defining the model and $\mathcal{F}(\theta)$ is the feasible set for the optimization problem. Then the optimal solution set is

$$\mathcal{F}^{opt}(\theta) := \arg \min_{x \in \mathcal{F}} f(x, \theta). \quad (11)$$

The same optimization problem can be done in similar yet varying models which gives multiple FOP and optimal solution sets:

$$FOP_i(\theta) := \min_x \{f_i(x, \theta) \mid x \in \mathcal{F}_i(\theta)\}, \quad \mathcal{F}_i^{opt}(\theta). \quad (12)$$

The inverse problem is then to find the estimate $\hat{\theta}$ of the parameters such that a known set of optimal decisions x_i^* fit as optimal solutions to the forward models $FOP_i(\hat{\theta})$. For a perfect estimate $\hat{\theta}$ the given x_i^* fit perfectly such that $x_i^* \subseteq \mathcal{F}_i^{opt}(\hat{\theta})$. The set of inverse-feasible estimates is then

$$\Theta_i^{inv}(x_i^*) := \{\theta \mid x_i^* \in \mathcal{F}_i^{opt}(\theta)\}. \quad (13)$$

If the inverse-feasible set is overdetermined, the inverse optimization problem need additional penalty functions to minimize over [4]. These would then act as model-specific objective functions, $p(\theta)$, penalizing bad estimates of θ .

For inverse optimization problems, [4] makes a distinction between Classical and Data-driven inverse optimization. Classical optimization applies $x_i^* \in \mathcal{F}_i^{opt}(\theta)$ as a constraint, that is to more strictly enforce perfect estimates:

$$IOP_C(x^*) := \min_{\theta} \{p(\theta) \mid \theta \in \Theta_i^{inv}(x_i^*), \theta \in \Theta\}. \quad (14)$$

Data-driven optimization applies $x_i^* \in \mathcal{F}_i^{opt}(\theta)$ as a loss function, that is being part of the objective function instead of the constraints:

$$IOP_D(x^*) := \min_{\theta} \{\kappa p(\theta) + \ell(x_i^*, \mathcal{F}_i^{opt}(\theta)) \mid \theta \in \Theta\}, \quad (15)$$

where κ is a weight, $p(\theta)$ is the penalty function, $\ell()$ the loss function and $i \in \{1, \dots, N\}$ are the samples of varying models.

To implement the constraint $\theta \in \Theta_i^{inv}(x_i^*)$ or the loss function $\ell(x_i^*, \mathcal{F}_i^{opt}(\theta))$, the optimality conditions can be used. Setting the KKT conditions as constraint or loss function couples the model parameters with the optimal choices. The conditions are valid both for the Forward Problem and the Inverse Problem, but for the latter as constraints instead of optimal criteria. The stationary condition can more easily be violated when applied in a loss function, since loosening the other optimality conditions would violate the feasible set of decision variables or the model itself.

F. The Markowitz Model

Recalling that the Markowitz Model has two objectives; to minimize risk while also maximizing return (mean-variance) of a portfolio of asset allocations, the portfolio selection problem can be written as $\max(\text{Return} - \text{Variance})$.

The fractional return of an asset can be modelled as $R = p_1/p_0$ where p_0 is the initial value and p_1 is the value after a time period. The rate of the asset's return is then $r = (p_1 - p_0)/p_0 = R - 1$. In the Markowitz model, the (fractional) return is modelled as a random quantity and instead its expected value $\mu = E[r]$ is used as mean. [1]

For a portfolio of n assets, μ becomes $\mu = [\mu_1, \dots, \mu_n]^T$ and the assets' respective weight factors $w = [w_1, \dots, w_n]^T$. By normalizing w so that $\sum_{i=1}^n w_i = \mathbf{1}^T w = 1$, the portfolio

is generalized for varying invested capital. An optional additional requirement $w_i \geq 0$ forbids short selling of assets (only long positions).

Total portfolio return rate is then $R = r^T w$ but modelled with its mean as $R := E[R] = \mu^T w$.

The covariance of the assets are modelled as

$$\Sigma = \text{cov}(r) = E[(r - \mu)^T(r - \mu)], \quad (16)$$

which gives the variance of the portfolio's total return rate as $V := \text{var}[R] = w^T \Sigma w$. [1]

To balance the model objectives according to the portfolio owner's preference, a multiplier parameter $\gamma \geq 0$ can be introduced to weight the return variance. This parameter can be interpreted as individual risk aversion of the portfolio owner: the higher γ the more risk averse.

Maximizing the expected return rate R can be reformulated as a minimum optimization problem since $\max_w R(w) = \min_w -R(w)$, and the optimization of the complete model then becomes $\min_w -(R - \gamma V)$.

More formally, an optimal portfolio w^* for an owner with the risk aversion γ and expecting the asset return rates μ is the solution to the minimization problem

$$\begin{aligned} \min_w \quad & f_0(w) := -\mu^T w + \frac{\gamma}{2} w^T \Sigma w \\ \text{s.t.} \quad & g_i(w) := (-w_i \leq 0), \quad i = 1, \dots, n \\ & h_i(w) := \mathbf{1}^T w - 1 = 0, \quad i = 1 \end{aligned} \quad (17)$$

where the first constraint is active if short selling of assets is forbidden.

The objective function of (17) is a quadratic function with the gradient

$$\nabla_w f_0(w) = -\mu^T + \frac{\gamma}{2}(\Sigma^T + \Sigma)w, \quad (18)$$

and the Hessian

$$\nabla_w^2 f_0(w) = \frac{\gamma}{2}(\Sigma^T + \Sigma). \quad (19)$$

Since any covariance matrix is symmetric ($\Sigma^T = \Sigma$) and positive semi-definite ($\Sigma \succeq 0$) the Hessian can be reduced to $\nabla_w^2 f_0(w) = \gamma \Sigma$ which yields that the Markowitz problem is a convex objective.

Furthermore, the constraints are affine functions, which in turn are convex [8]. Hence, the Markowitz model has a convex objective function and a feasible set that is convex:

$$\begin{aligned} f_0(w) : \mathcal{C} &\rightarrow \mathbb{R} \\ w \in \mathcal{F} &= \{w \in \mathbb{R}^n \mid \mathbf{1}^T w = 1, (w_i \geq 0 \ i = 1, \dots, n)\} \subseteq \mathcal{C}. \end{aligned}$$

G. Optimality Conditions for the Markowitz Model

Revisiting the KKT conditions presented in subsection II-D, conditions can be stated for any optimal portfolio in two separate cases: when short selling of assets are allowed and not allowed. Both of them require the primal feasibility condition $\mathbf{1}^T w^* = 1$.

In the case where short selling is allowed, the stationary condition becomes

$$\begin{aligned} \nabla_w f_0(w^*) + \sum_{i=1}^1 \nu_i \nabla_w h_i(w^*) &= 0 \\ \nabla_w (-\mu^T w^* + \frac{\gamma}{2} w^{*T} \Sigma w^*) + \nu \nabla_w (\mathbf{1}^T w^* - 1) &= 0 \\ -\mu + \gamma \Sigma w^* + \nu \mathbf{1} &= 0. \end{aligned} \quad (20)$$

In the case where only long positions are allowed, a second constraint ($w_i \geq 0$) is added. The stationary condition becomes

$$\begin{aligned} \nabla_w f_0(w^*) + \sum_{i=1}^n \lambda_i \nabla_w g_i(w^*) + \sum_{i=1}^1 \nu_i \nabla_w h_i(w^*) &= 0 \\ \nabla_w (-\mu^T w^* + \frac{\gamma}{2} w^{*T} \Sigma w^*) + \sum_{i=1}^n \lambda_i \nabla_w (-w_i^*) &+ \nu \nabla_w (\mathbf{1}^T w^* - 1) = 0 \\ -\mu + \gamma \Sigma w^* + \sum_{i=1}^n (-\lambda_i e_i) + \nu \mathbf{1} &= 0 \\ -\mu + \gamma \Sigma w^* - \lambda + \nu \mathbf{1} &= 0. \end{aligned} \quad (21)$$

Furthermore, the dual feasibility condition $\lambda_i \geq 0$ and the complementary slackness condition $-\lambda_i w_i^* = 0$ must also be satisfied for the optimal portfolio.

H. Inverse Optimization of the Markowitz Model

The optimal portfolio model in (17) is characterized by the parameter set $\theta = \{\mu, \gamma, \Sigma\}$ and the decision variable w with optimal solution set $\mathcal{F}_i^{\text{opt}}(\theta) = \{w_i^*\}$.

In the case when short selling is allowed, the stationary KKT condition is $-\mu^T + \gamma \Sigma w^* + \nu \mathbf{1}^T = 0$. A classical inverse optimization problem estimating the parameters θ (where w^* is a known constant) would then be

$$\begin{aligned} \min_{\mu, \gamma, \Sigma} \quad & p(\mu, \gamma, \Sigma) \\ \text{s.t.} \quad & -\gamma_i \leq 0, \quad i = 1, \dots, j \\ & -\mu^T + \gamma \Sigma w^* + \nu \mathbf{1}^T = 0, \end{aligned} \quad (22)$$

where $p(\theta)$ is a convex penalty function and $\nu \in \mathbb{R}$. Similarly, with $\lambda \in \mathbb{R}^n$ the case when short selling is forbidden the classical inverse optimization problem would be

$$\begin{aligned} \min_{\mu, \gamma, \Sigma} \quad & p(\mu, \gamma, \Sigma) \\ \text{s.t.} \quad & -\gamma, -\mu_i, \lambda_i \leq 0, \quad i = 1, \dots, n \\ & -\lambda_i w_i^* = 0, \quad i = 1, \dots, n \\ & -\mu^T + \gamma \Sigma w^* - \lambda + \nu \mathbf{1}^T = 0. \end{aligned} \quad (23)$$

Let $\kappa \geq 0, \kappa \in \mathbb{R}$ be a weight parameter. A corresponding Data-driven IOP would then for the short selling allowed-case be

$$\begin{aligned} \min_{\mu, \gamma, \Sigma} \quad & \kappa p(\mu, \gamma, \Sigma) + \|\mu^T + \gamma \Sigma w^* + \nu \mathbf{1}^T\|_2 \\ \text{s.t.} \quad & -\gamma_i \leq 0, \quad i = 1, \dots, j, \end{aligned} \quad (24)$$

and for the short selling forbidden-case be

$$\begin{aligned} \min_{\mu, \gamma, \Sigma} \quad & \kappa p(\mu, \gamma, \Sigma) + \|\mu^T + \gamma \Sigma w^* - \lambda + \nu \mathbf{1}^T\|_2 \\ \text{s.t.} \quad & -\gamma, -\mu_i, \lambda_i \leq 0, \quad i = 1, \dots, n \\ & -\lambda_i w_i^* = 0, \quad i = 1, \dots, n. \end{aligned} \quad (25)$$

III. METHOD

A. Implementing the Markowitz Model

The central parameters defining a market state of the Markowitz Model are $\mu \in \mathbb{R}^n$ and $\Sigma \in \mathbb{R}^{n \times n}$, representing the expected return of n assets and their covariance (which is indirectly defined by μ in (16)). Based on this believed market state, an agent with risk aversion $\gamma \in \mathbb{R}$ can solve the Forward Optimization in equation (17) to generate their individual optimal portfolio $w^* \in \mathbb{R}^n$ with n assets.

Throughout the report, artificial data will be used to simulate asset and agent data. In the following simulations and tests, randomized values have been applied as proxies for μ and Σ . μ was modelled as

$$\mu = \{|\mu_i| \mid \mu_i \sim \mathcal{N}(0, 1), i = 1, \dots, n\}, \quad (26)$$

and Σ was modelled as

$$\Sigma = \{V^T V \mid V_{i,k} \sim \mathcal{N}(0, 1), V \in \mathbb{R}^{n \times n}, i, k = 1, \dots, n\}, \quad (27)$$

where $V \in \mathbb{R}^{n \times n}$ acts as proxy for $(r - \mu)$ for n assets. All entries of V are independent from each other.

Extending the model to comply with j multiple agents acting on the same believed market state μ of n assets, the extended parameters can be written as

$$\begin{aligned} \mu' &= (\mu, \mu, \dots, \mu) \in \mathbb{R}^{n \times j}, \quad \gamma' = (\gamma_1, \gamma_2, \dots, \gamma_j) \in \mathbb{R}^j \\ w' &= (w_1, w_2, \dots, w_j) \in \mathbb{R}^{n \times j}, \quad \Sigma' = \Sigma \in \mathbb{R}^{n \times n} \end{aligned} \quad (28)$$

In this report, the parameters will be written as μ, γ, w, Σ for both the simple or the extended form. The intended form will be indicated with support of sub-scripted indexes, assigned dimension or by the context.

For the implementation, the respective risk aversions of j agents were modelled as

$$\gamma = \{\gamma_i \in [l, h] \mid \log_{10} |\gamma_i - \gamma_{i+1}| = d, i = 1, \dots, j\} \quad (29)$$

where l and h is the lower and higher bound respectively of the logarithmic evenly distributed (with distance d) γ_i .

B. Parameter Estimation

The complete model of the market state, agent behaviors and their asset portfolios sum up as $\theta = \{\mu, \Sigma, \gamma, w\}$. While the forward optimization problem has the portfolio weights w as the decision variable and μ, Σ and γ as known constant, the inverse optimization problem examined in this work has μ and γ as decision variables and w^* and Σ as known constants. Again, choosing only these two parameters to be estimated is due to them being of most interest and to limit the scope of the work. Under these assumptions the KKT conditions becomes affine, which simplifies the inverse optimization.

The Inverse Optimization Problem can then be written as

$$IOP(w^*) := \min_{\mu, \gamma} \{p(\theta) \mid \{\mu, \gamma\} \in \Theta^{inv}(w^*)\} \quad (30)$$

and optimal solution is then the estimate pair

$$(\hat{\mu}, \hat{\gamma}) = \arg \min_{\mu, \gamma} \{p(\theta) \mid \{\mu, \gamma\} \in \Theta^{inv}(w^*)\}. \quad (31)$$

Since the IOP has a larger degree of freedom than the FOP and acts as an overdetermined system, further constraining actions were needed. This was addressed by implementing three model-specific penalty functions.

Assuming all agents operate on the same market and that their believed knowledge of the market state is (somewhat) common, the estimate $\hat{\mu}$ should be equal among the agents. With $\hat{\mu} \in \mathbb{R}^{n \times j}$ and $\hat{\mu}_i \in \mathbb{R}^{n \times 1}$ this gives that $\hat{\mu}_i - \hat{\mu}_k \in \mathbb{R}^{n \times 1}$ represents the difference between the two estimates of expected return by agent i and k .

This multidimensional difference can be evaluated as a scalar value by summarizing the Euclidean norm of the column differences of matrix $\hat{\mu}$. The first penalty function evaluates the differences between estimates of an agent and estimates of its neighbors:

$$p_C(\mu) = \sum_{i=1}^j \|\mu_i - \mu_{i-1}\|_2 + \|\mu_i - \mu_{i+1}\|_2 \quad (32)$$

An alternative evaluation is to summarize the differences element-wise. The second penalty function evaluates the element-wise difference between estimates of neighboring agents' expected return $\hat{\mu}_{i,k}$:

$$p_E(\mu) = \sum_{k=1}^n \sum_{i=1}^j |\mu_{i,k} - \mu_{i,k-1}| + |\mu_{i,k} - \mu_{i,k+1}| \quad (33)$$

Although identical estimates of agents' expected return μ are convenient, the ultimate aim is a perfect estimate such that $\hat{\mu} = \mu$. While not knowing true μ , a reasonable assumption would be that μ does not deviate significantly from its previous values. For example, a mean value $\bar{\mu}$ of historical values during a not-too-long time period could suffice as estimation of true μ . Furthermore, the existing deviation of the current value on an asset ($\mu_{j,n}$) from their past value needs to be related to the covariance Σ between the assets. A penalty function based on these assumptions can be written for a single agent as

$$p(\mu_i) = (\mu_i - \bar{\mu})^T \Sigma^{-1} (\mu_i - \bar{\mu}) \quad (34)$$

Extended to j multiple agents, the third penalty function becomes

$$p_H(\mu) = \sum_{i=1}^j \left((\mu - \bar{\mu})^T \Sigma^{-1} (\mu - \bar{\mu}) \right)_{i,i} \quad (35)$$

where $\mu \in \mathbb{R}^{n \times j}$ and $\bar{\mu} \in \mathbb{R}^{n \times 1}$. Σ^{-1} exists for $\Sigma \succeq 0$.

Since records of historical values of μ were not available, $\bar{\mu}$ was modelled using Gaussian distributions as a proxy:

$$\bar{\mu} = \{\bar{\mu}_i \mid \bar{\mu}_i \sim \mathcal{N}(\mu_i, \frac{1}{N} \Sigma), i = 1, \dots, n\} \quad (36)$$

The estimates of $(\hat{\mu}, \hat{\gamma})$ presented below were derived using the penalty functions $p_C(\mu)$, $p_E(\mu)$, $p_H(\mu)$ and $\bar{\mu}$ defined and modelled as above.

C. Software Implementation

The artificial implementation of the Markowitz Model and estimations as described above were realized in software using Python as the programming language and the software libraries

NumPy [10], SciPy [11], scikit-learn [12] and CVXPY [13], [14]. Additionally, figures were extracted using the library Matplotlib [15]. Given that the optimization problem in the Markowitz model is convex, the applied optimization solvers imported from the CVXPY library are central for the performance of the estimations presented in this report.

CVXPY is an open-source and domain-specific (python-embedded) library intended for solving convex optimization problems. It has a relatively rich API and by default chooses a solver (algorithm) best suited for the characteristics of the problem. CVXPY is a well-known and established library judged to both converge with high accuracy at a small number of iterations and to be robust in terms of data scaling.

When solving the Markowitz optimization problem for optimal portfolios, CVXPY applied OSQP [16] as the solver. OSQP, Operator Splitting Quadratic Program, solves convex quadratic problems written as $\min_x \frac{1}{2}x^T Px + q^T x$ (which is well-suited for the Markowitz problem $\min_w \frac{\gamma}{2}w^T \Sigma w - \mu^T w$). OSQP in turn, applies an ADMM (Alternating Direction Method of Multipliers) solver algorithm [17].

When solving the inverse optimization problem, CVXPY applied ECOS [18] as solver. ECOS, Embedded Conic Solver, solves second-order cone problems are written as

$$\begin{aligned} \min_x \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & Gx + s = h, \quad s \in \mathcal{K} \end{aligned} \quad (37)$$

where $c \in \mathbb{R}^n$, $b \in \mathbb{R}^p$, $A \in \mathbb{R}^{p \times n}$, $G \in \mathbb{R}^{M \times n}$, $h \in \mathbb{R}^M$ and the order of the cone \mathcal{K} is M . ECOS in turn, applies interior-point solver algorithms.

D. Application: Market State Sorting

How satisfying successful parameter estimations may be to a mathematician, practical applications of it give additional joy to an engineer. The ability to discover previously unknown information from a data sample offers answers to several interesting questions. In the case of asset portfolio trading, information on the portfolio agents is of high relevance and interest. What did a portfolio owner know – or believed to know – when deciding their portfolio?

Agent's knowledge of the market may not be completely common or equal to others, such as situations where an agent has insider or non-public information about some assets' expected return. Their optimal portfolio would then differ according to their believed knowledge. Detecting this divergent behavior could be of interest to expose eventual violations of confidentiality or financial regulation.

Agents having different beliefs about the market can be said to operate in different versions of the market. Denoting an agent's belief of the market as market state μ^i , a group of agents can be grouped in M clusters of similar market states as

$$\mu^i = \{\mu^s \mid i \in \Omega^s, \quad s = 1, \dots, M, \quad i = 1, \dots, j\} \quad (38)$$

where Ω^s is the set of agents in cluster s for M clusters of the market. This gives $\text{card}(\Omega) = j$ and $\bigcup_{i=1}^M \Omega^s = \Omega$, along

with the condition that an agent can only belong to one market state.

This work demonstrates these anomalies with an experiment with two market states μ^1 and μ^2 . The question to answer was *Which agents act in market state μ^1 and which in μ^2 ?*

The experiment modelled the market states as two independent markets defined in (26). The group Ω^1 of agents operating on μ^1 were assumed to be larger than the group Ω^2 for μ^2 . Agents of Ω^2 were then randomly placed among agents of Ω^1 . The parameters μ, γ, Σ, w were extended similarly to the case with a single market, but now with μ^2 and w^2 corresponding to Ω^2 -agents hidden in a larger group. γ was allocated as before irrespective to the agent's market state.

This enabled the estimation to be performed regardless of eventual different market versions. The estimates were then used in a sorting algorithm, grouping the agents back into two groups based on similar market estimates $\hat{\mu}$. The sorting algorithm combined the scikit-learn functions PCA and AgglomerativeClustering to cluster agents' index in two groups. To simplify the clustering, PCA projected the multidimensional matrix $\hat{\mu}$ into a two-dimensional data set using Singular Value Decomposition. AgglomerativeClustering then clustered the agents in two groups using bottom-up hierarchical sorting until two clusters were reached.

The proposed grouping of agents ($\hat{\Omega}^1, \hat{\Omega}^2$) could then be compared with the true groups (Ω^1, Ω^2).

E. Estimation Correctness

Despite different combinations of the penalty functions $p_C(\mu)$, $p_E(\mu)$, $p_H(\mu)$, neither classical or data-driven inverse optimizations will likely yield perfect estimates $(\hat{\mu}, \hat{\gamma}) = (\mu, \gamma)$. This is due to multiple factors, such as the estimation being done on an overdetermined system or the lack of penalty functions regulating $\hat{\gamma}$ and dual variables λ, ν . Thus, the estimates will be incorrect and a relevant question becomes *how* incorrect.

An intuitive approach to assessing the correctness of the estimation is the deviation of the estimate from its true value. The estimate deviation for $\hat{\mu} \in \mathbb{R}^{n \times j}$ and $\hat{\gamma} \in \mathbb{R}^j$ using the Frobenius and Euclidean norm can be written as

$$D_\mu(\hat{\mu}) = \|\hat{\mu} - \mu\|_F, \quad D_\gamma(\hat{\gamma}) = \|\hat{\gamma} - \gamma\|_2. \quad (39)$$

The calculated scores are useful for comparing and optimizing (such as weighting) the choice of penalty functions and the type of inverse optimization. However, a second question arises: are the estimates correct *enough*?

This is answered by examining possible applications of the estimation. In this work, the market state sorting problem acts as a validation of the estimates' quality. The performance for the agent sorting was then examined for different number of n assets, j agents and $\text{card}(\Omega^2)$.

F. Estimation Robustness

To account for the considerable simplification of artificial data, noise was added to the estimation model. The noise $v \in \mathbb{R}^{n \times j}$ added to optimal portfolios resulted in sub-optimal portfolios w^v which both is a more realistic model

of asset portfolios and enables robustness assessment of the estimations. Gaussian noise was used as a proxy for noise, which can be written as

$$v = \{v_{i,k} \mid v_{i,k} \sim \mathcal{N}(\phi, \sigma), i = 1, \dots, j, k = 1, \dots, n\}, \quad (40)$$

where ϕ is the mean value and σ the standard deviation.

Given that $w_i^* \in [-1, 1]$ (or $w_i^* \in [0, 1]$ when short selling forbidden), noise with $\phi = 0$ and $\sigma \in [0, 1]$ were judged appropriate to test robustness. The market state sorting problem was then implemented with the sub-optimal portfolios $w^v = w^* + v$ and the correctness rate $\text{card}(\hat{\Omega}^2)/\text{card}(\Omega^2)$ was examined.

IV. RESULTS

Below are results relevant to exemplify the parameter estimation and its applications presented. For the results presented in this section, short selling was allowed and $N = 1000$ was used in equation (36) to model historical values $\bar{\mu}$ for the penalty function $p_H(\mu)$.

Results concerning the parameter estimation are presented in IV-A. Results regarding the market state sorting problem and model robustness are presented in IV-B.

A. Parameter Estimation Accuracy

To give an overview of the resulting estimates, an excerpt of true values and estimates of μ and γ are presented in Table I and II. Results for penalty function p_C are presented in Tables I and II. The results when no penalty function was used (only stationary KKT conditions as objective function) are presented in Tables III and IV. Number of agents j was set to 500 and number of assets n set to 55. The first, last and two neighboring agents are displayed. Similarly the first, last and two adjacent assets are shown.

TABLE I
EXAMPLES OF VALUES FOR μ AND $\hat{\mu}$ WITH $p_C(\mu)$.

True μ				
	Agent 1	Agent 250	Agent 251	Agent 500
Asset 1	1.6243	1.6243	1.6243	1.6243
Asset 27	0.12289	0.12289	0.12289	0.12289
Asset 28	0.93577	0.93577	0.93577	0.93577
Asset 55	0.20889	0.20889	0.20889	0.20889
Estimated $\hat{\mu}$				
	Agent 1	Agent 250	Agent 251	Agent 500
Asset 1	1.0219	1.0219	1.0219	1.0219
Asset 27	1.0219	1.0219	1.0219	1.0219
Asset 28	1.0219	1.0219	1.0219	1.0219
Asset 55	1.0219	1.0219	1.0219	1.0219

The deviation of the estimates from their true values, when applying the penalty functions separately, are presented in Table V. Number of agents j were set to 50, 500 and 1000 for $D_\mu(\hat{\mu})$ and 50, 250 and 500 for $D_\gamma(\hat{\gamma})$. Number of assets n was set to 150. The summarized deviations D_μ and D_γ were then normalized by division of respective number of agents.

TABLE II
EXAMPLES OF VALUES FOR γ AND $\hat{\gamma}$ WITH p_C

Agent index	1	250	251	500
True γ	0.010	0.318	0.322	10.00
Estimated $\hat{\gamma}$ ($\times 10^{-12}$)	1.60	8.32	8.38	30.1

TABLE III
EXAMPLES OF VALUES FOR $\hat{\mu}$ WITH DATA-DRIVEN INVERSE OPTIMIZATION ONLY

Estimated $\hat{\mu}$				
	Agent 1	Agent 250	Agent 251	Agent 500
Asset 1	1.2010	1.2352	1.2349	1.0855
Asset 27	1.1353	1.0800	1.0800	1.0097
Asset 28	1.0376	1.0697	1.0687	1.0507
Asset 55	0.9766	1.0471	1.0475	1.0140

TABLE IV
EXAMPLES OF VALUES FOR $\hat{\gamma}$ WITH DATA-DRIVEN INVERSE OPTIMIZATION ONLY

Agent index	1	250	251	500
Estimated $\hat{\gamma}$	9.27×10^{-6}	0.297	0.299	1.247

TABLE V
TOTAL DEVIATION FROM TRUE μ AND γ

$D_\mu(\hat{\mu})/j$ with $n = 55$			
j	$p_C(\mu)$	$p_E(\mu)$	$p_H(\mu)$
50	0.58998	0.58998	0.58998
250	0.26127	0.26127	0.26127
500	0.18561	0.18561	0.18561
$D_\gamma(\hat{\gamma})/j$ with $n = 55$			
j	$p_C(\mu)$	$p_E(\mu)$	$p_H(\mu)$
50	0.38140	0.38140	0.38140
250	0.16328	0.16328	0.16328
500	0.11452	0.11452	0.11452

B. Performance of Market State Sorting

The market state sorting problem was solved for a varying number of agents of which 10% belonged to a second market state μ^2 . The number of agents j was set to increment from 50 to 500 with steps of 50. The number of assets was set to 55. The performance of the sorting algorithm using estimations with penalty functions $p_C(\mu)$, $p_E(\mu)$ and $p_H(\mu)$ applied separately are shown in Table VI.

The table details how many agents belonged to the second market state, Ω^2 , how many agents the algorithms identified as belonging to the second market state, $\hat{\Omega}^2$, how many of these that were correctly or wrongly identified, and finally how many agents of Ω^2 the algorithm missed to identify.

Due to the algorithm performing completely correct sorting for higher number of agents ($j = [150, 450]$), these data has been left out.

For the robustness test, the parameter estimation was done

TABLE VI
MARKET STATE SORTING PERFORMANCE

$p_C(\mu)$					
j	$\text{card}(\Omega^2)$	$\text{card}(\hat{\Omega}^2)$	Correct	Wrong	Missing
50	5	15	0	15	5
100	10	10	10	0	0
150	15	15	15	0	0
500	50	50	50	0	0
$p_E(\mu)$					
j	$\text{card}(\Omega^2)$	$\text{card}(\hat{\Omega}^2)$	Correct	Wrong	Missing
50	5	14	0	14	5
100	10	47	10	37	0
150	15	15	15	0	0
500	50	50	50	0	0
$p_H(\mu)$					
j	$\text{card}(\Omega^2)$	$\text{card}(\hat{\Omega}^2)$	Correct	Wrong	Missing
50	5	5	5	0	0
100	10	10	10	0	0
150	15	15	15	0	0
500	50	50	50	0	0

with classical inverse optimization using all three penalty functions combined with equal weights. Fig. 1 presents the rate of correctness (correctly identified compared to the total number of second market state agents), of the market state sorting algorithm as Gaussian noise added to the optimal portfolios increases. The noise increased in respect of its variance, by increasing the standard deviation of the noise proxy such that $\sigma = [0, 0.5]$. The figure includes 30 simulations for 500 agents and 55 assets, with the black curve denoting the mean performance.

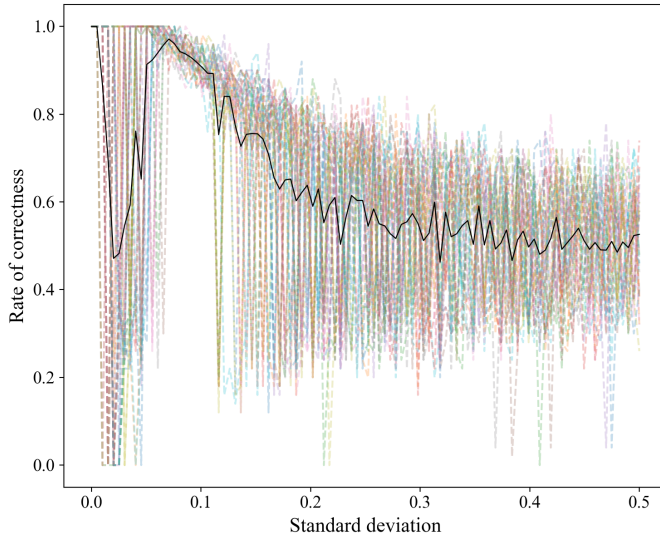


Fig. 1. Sorting performance for increasingly sub-optimal portfolios.

V. DISCUSSION

A. Limitations

The primary aim of the work in this report is to demonstrate the use of inverse optimization. The model used for this has for convenience been simplified in several ways. The Markowitz model is a relatively primitive theory only targeting the expected return rate and its variance of a portfolio. More complex models could take into account other relevant aspects for portfolio optimization.

A major simplification of the portfolio operations done in this work is the artificial data. The values of μ is the standard normal distribution, and the covariance of the assets Σ are completely uncorrelated to $E[r] = \mu$ since it is a separate random sample of the standard normal distribution. Furthermore, γ is a personal parameter for any portfolio owner which is difficult to realistically model. In this model, however, the γ_i are spread in sorted order among the agents. Thus, an unrealistic and potentially helpful relation originates between a portfolio's w_i placement in w and the corresponding portfolio owner's γ_i .

In the real-world world additional reasonable assumptions of the agents could potentially be made of the optimal portfolios. With knowledge of the portfolio's owner, examples of these could be whether short selling is allowed, relation to other agents' risk aversion or preference in specific kinds of assets (such as technology, petrol, or nationality). The inverse optimization problem could then implement additional constraints which would give a more accurate estimation. No case-specific assumption has been included in this work which limits the estimations to general portfolios.

The software implementation also has limitations in terms of speed and capacity. The used library functions for optimization and sorting are well-qualified for the problem and suitable for scaling, while the overall program structure and memory management become a challenge for large portfolios or many agents. In this work, a relatively simple yet efficient implementation was done.

For solving the Markowitz model optimization (generation of optimal portfolios) it was observed that the program failed for $n > 170$ assets, while scaling the number of agents only increased the time required. For the inverse optimization (estimating μ and γ) – already limited to $n \leq 170$ – a larger number of agents also increased the time required. This is explained by the increased number of iterations done but also the more computer-heavy matrix operations performed. As a reference, $j = 4000$ agents were executed in approximately 90 minutes. These issues could potentially be avoided or limited by using more efficient data structures and sorting algorithms, like scalable optimizers that exploit the problem at hand.

B. Accuracy of Estimation

As can be seen in Table I, all estimates are identical for all assets and all agents. This was the case for all penalty functions, which is why only the results of the column-wise penalty function are shown. The suspected reason as to why the estimates are so poor can be seen in Table II. Here, all estimates of gamma are practically zero. Recalling the

optimality conditions $-\mu^T + \gamma \Sigma w^* + \nu \mathbf{1}^T = 0$, one can understand why setting gamma to very small values will result in poor estimates: the information from the different optimal portfolios w^* becomes neglected and unused. It is suspected that these bad values of gamma are the result of not having a penalty function that addresses or regulates gamma. Only penalty functions for μ were developed, as it was believed that they would suffice. However, a simplified inverse optimization using true γ values and only estimating the parameter μ , showed that the application of the defined penalty functions had a positive impact on regulating the estimates.

There were attempts made to lessen the influence of the penalty function by weighing it less in the loss function. The results showed that the best estimates were attained when the penalty function was set to zero, that is only the stationary KKT condition were set as a loss functions in the objective function. These results are presented in Tables III and IV.

As can be seen in Table III, the estimates of μ are not very good in absolute terms. Looking at asset 1, the true values of μ are between 25% and 50% larger than that of the estimated values. Looking at assets 27, 28 and 55, the opposite is true. Here the estimated value of μ is sometimes as much as 10 times higher, as is the case in asset 27 for agent 1.

Traces of the penalty functions promoting neighbors having similar $\hat{\mu}$ can be seen by looking at agents 250 and 251. The difference between $\hat{\mu}_{250}$ and $\hat{\mu}_{251}$ is comparatively much smaller than with $\hat{\mu}_1$ or $\hat{\mu}_{500}$.

A similar case can be made for the estimates of γ , which can be seen in IV. Here, the estimates are even worse. The estimate $\hat{\gamma}$ for agent 1 differs from that of the true value with a factor of 10^{-3} , but it manages to make decent estimates of agents 27 and 28. This might be a consequence of the model promoting agents to have a similar $\hat{\mu}$ as their neighbor, and the optimization algorithms set their $\hat{\gamma}$ to be similar as well.

This may be due to neighboring optimal portfolios being generated with similar value on γ . This could mean that for the KKT stationary condition to hold, when μ is penalized to become equal, the estimated γ would also need to be similar among its neighbours. This spill-over effects from the penalty functions could also be an explanation for the inability to estimate high values of γ , observed in Table IV by comparing $\hat{\gamma}_{500}$ and $\hat{\gamma}_{500}$.

The assumed major reason why the estimates of γ are poor is that no penalty function promoted the model to calculate accurate $\hat{\gamma}$. Still, even though the estimates are poor in absolute terms, the model manages to identify the agents' risk aversion relative to each other ($\hat{\gamma}_1 < \hat{\gamma}_2 < \dots < \hat{\gamma}_i$).

Table V shows that an increase in the number of agents relates to the model performing better (a lower score is better).

The introduction of noise can be seen as a deviation from the optimal portfolios, on which the model relies. As can be seen in Fig. 1, the model successfully sorted approximately 80% of agents when $\sigma = 0.1$. Considering that the sum of the weighted portfolio is 1 and the number of assets was 55, the mean weight per asset in w^* is 0.018. From this view, noise with $\sigma = 0.1$ would in general impose relatively great noise. When noise is first introduced, a large dip in the amount of correctly identified agents can be seen. Possible explanations

for this could be a flaw in the software implementation, such as the inverse optimization solver in CVXPY is falsely satisfied when noise is introduced.

Another observation is the non-varying performance when $\sigma = [0.06, 0.1]$ where the sorting algorithm seems highly robust. This could be seen as a hint that it is an implementation flaw causing the instant dip for $\sigma = [0, 0.01]$.

The share of correctly sorted agents converges towards 50%, which is to be expected considering that if the sorting algorithm can't find a correlation between data points, it will still sort agents into two clusters but arbitrarily. This will result in 50% of the agents belonging to the second market ending up in the by the algorithm identified group.

C. Future Work and Outlook

In this report the Markowitz model was implemented with no restriction on short selling of assets. To assess the model's generality, it could be of relevance to do the corresponding implementation with short selling forbidden. Furthermore, to improve the model, the penalty functions could be examined and eventually redesigned. What possible penalty functions exist for regulating γ that could improve the estimation? These questions are suitable for future work.

This work includes a single application of the estimated parameters. Of course, several other applications would be possible. For example, situations where the only interest is to find the dominating believed market state (and thus potentially find the true belief due to the law of large numbers) or to identify divergent portfolio agents disregarding their market state. For this, the median of the estimates might be a useful proxy for modeling a dominating market belief.

Another important quality of a model is its applicability to real-world data. The available model could be tested on real asset records to evaluate its potential for real-world use.

Looking beyond portfolio selection, the use of inverse optimization for parameter estimation should be possible in many other disciplines. The requirements for a situation to be applicable would be a large amount of optimal choices available, the situation itself being possible to describe mathematically and the existence of valid assumptions suitable for optimization regulation. Examples of this could be in image or sound processing, where the scientific knowledge is mature and assumptions can be made on the occurrence of specific frequencies or pixel relations.

VI. CONCLUSION

The work presented in this report concludes that inverse optimization is applicable for parameter estimation of convex optimization objectives such as the Markowitz model. Three different regulating functions for inverse optimization were identified and implemented. However, to show the full potential and accuracy of inverse optimization more work is needed. The authors do not see a prevalent reason why it would not be possible to accurately regulate the estimates, but due to time constraints were not able to complete a successful estimate regulation.

Furthermore, the work demonstrates a somewhat successful implementation of a parameter estimation and evaluates the estimation's correctness, applicability and robustness.

The analysis made is that the performance of the estimation depends on both the qualification of the model and on the amount of optimal decisions available. Based on the work showing a positive relation with better estimation for larger amount of data, it can be recommended to maximize the availability of data given eventual constraints imposed by limited implementation capacity.

ACKNOWLEDGMENT

The authors would like to thank their supervisor Jacob Lindbäck for his continuous guidance, availability and engagement in the work. We wish him the best of luck in his academic research and his quest for iPad chargers.

REFERENCES

- [1] H. Markowitz, "Portfolio selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952. [Online]. Available: <http://www.jstor.org/stable/2975974>
- [2] T. C. Chan, T. Lee, and D. Terekhov, "Inverse optimization: Closed-form solutions, geometry, and goodness of fit," *Management science*, vol. 65, no. 3, pp. 1115–1135, 2019.
- [3] R. K. Ahuja and J. B. Orlin, "Inverse optimization," *Operations Research*, vol. 49, no. 5, pp. 771–783, 2001. [Online]. Available: <http://www.jstor.org/stable/3088574>
- [4] T. C. Y. Chan, R. Mahmood, and I. Y. Zhu, "Inverse optimization: Theory and applications," 2021. [Online]. Available: <https://arxiv.org/abs/2109.03920>
- [5] J. Y.-M. Li, "Inverse optimization of convex risk functions," *Management science*, vol. 67, no. 11, pp. 7113–7141, 2021.
- [6] G. Iyengar and W. Kang, "Inverse conic programming with applications," *Operations research letters*, vol. 33, no. 3, pp. 319–330, 2005.
- [7] R. Mattila, I. Lourenco, C. R. Rojas, V. Krishnamurthy, and B. Wahlberg, "Estimating private beliefs of bayesian agents based on observed decisions," *IEEE Control Systems Letters*, vol. 3, no. 3, pp. 523–528, 2019.
- [8] S. Boyd and L. Vandenberghe, *Convex Optimization*. The Edinburgh Building, Cambridge, CB2 8RU, UK: Cambridge University Press, 2004. [Online]. Available: https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf
- [9] A. Sasane and K. Svanberg, *Optimization*. Stockholm: Department of Mathematics, Royal Institute of Technology, Stockholm, 2013. [Online]. Available: <https://personal.lse.ac.uk/sasane/Optimization.pdf>
- [10] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. [Online]. Available: <https://doi.org/10.1038/s41586-020-2649-2>
- [11] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, I. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.
- [12] 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.
- [13] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.
- [14] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, "A rewriting system for convex optimization problems," *Journal of Control and Decision*, vol. 5, no. 1, pp. 42–60, 2018.
- [15] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.
- [16] 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>
- [17] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011. [Online]. Available: <http://dx.doi.org/10.1561/22000000016>
- [18] A. Domahidi, E. Chu, and S. Boyd, "ECOS: An SOCP solver for embedded systems," in *European Control Conference (ECC)*, 2013, pp. 3071–3076.

CONTEXT C – PART II

LEARNING IN DYNAMICAL SYSTEMS

POPULAR DESCRIPTION

How long until your dog loses its job?

Choosing the self scanning section, using self driving vehicles or talking to chatbots in tech support – replacing our fellow humans is not something we think twice about in the face of cheaper and more convenient solutions.

But are we ready to replace our best friend, the dog? What if our furry friend wouldn't pee on the rug, bark at the neighbors or need to take walks? Let us introduce the AI dog, mankind's new best friend.

You probably wouldn't be too keen on replacing your dog, but what if your dog's best trick wasn't to sit on command but the ability to keep learning new things like cleaning the house, unloading the dishwasher or even doing your taxes? Talk about teaching an old dog new tricks.

But what keeps us skeptical about replacing our real dog with a robot? It is that we don't think they will be able to give us the same emotional connection that we have with living creatures. In the future, AI will probably be able to develop dog-like emotions, and perhaps your AI dog will actually be your new best friend

The robot dog is just one example of the possibilities the future of AI gives us. You can already see AI changing our daily lives: checkout-less stores have already opened in London, meaning you just need to pick the items you want and walk out of the store. Nearly every workplace that decides to implement AI will see an immediate improvement in efficiency and new possibilities will continuously present themselves, just like with the robot dog.

Self-driving cars, checkout-less stores and advanced virtual assistants are already emerging. While the robot dog might be decades away, we will see that in the coming years an increased amount of AI:s implemented in our daily lives and in the meantime we will have to settle for our ordinary old furry friends.

SUMMARY OF PROJECT RESULTS

In recent years artificial intelligence (AI) has been gaining a lot of attention in its ability to learn how to perform complex tasks by analyzing huge amounts of data. This is possible because of the accessibility to powerful computing in recent years. The problem is that more often than not the amount of data that is needed for an AI to learn by itself is not always available. A method that has been used to counter this problem is reinforcement learning (RL). The difference between this method and the more well-known AI methods is that RL learns by trial and error and does not need any prior knowledge about the system. This is especially useful for dynamical systems i.e. the behavior of the system changes depending on what has happened before. A classic example of a dynamical system is chess.

In project C2a we explored the possibilities of RL by implementing an algorithm on a video game with the goal of getting the highest score possible, and investigating if the algorithm can play better than a human player. The algorithm is not given any prior experience or information about the game, but can through many playthroughs learn a strategy to maximize the game score. A stepping-stone was implementing the algorithm on the classical control theory problem cartpole, which is an inverted pendulum. Future developments of this project could be to implement other deep reinforcement learning algorithms to see if they perform better, and incorporating the algorithm on more complex environments.

In project C2b, RL was applied to the card game “Limit Texas Hold’em”, a popular variant of the game poker, with the aim to create an agent that could play on a human level. To facilitate this, two different algorithms were implemented, Deep Q-learning and Deep Monte Carlo. To achieve the goal of the project, the different parameters of the algorithm have been tweaked to make it learn faster and more effectively. Whilst human performance was not achieved, the algorithms learned to play the game at a reasonable level. For future extensions, it would be prudent to implement a better self-play algorithm, where the RL algorithm will play against itself, as this would allow it to achieve a greater performance.

In project C3, the goal was to use RL to train multiple robots in a warehouse to cooperate in transporting boxes while avoiding collisions. The warehouse environment was constructed by the group as a simulation where boxes, their destinations and all the robots will roam. Each robot observes the environment and takes actions it finds fit to achieve its goal of transporting boxes and avoiding crashing. The algorithm used to achieve this was Deep Q-learning. The further goal was to construct the deep algorithm to efficiently scale training time for more complex environments. The objective for future studies could be to optimize performance and experiment with increasing the complexity in other ways. While this project has mainly been a study of scalability in Deep Q-learning and how it performs, a comparison should be made with scalability in other algorithms.

In conclusion, RL can be used to solve a wide range of complex tasks in dynamical systems, and it will be exciting to see how reinforcement learning will be further developed and used in the future. However, the big question that needs to be discussed is which fields should not be solved with reinforcement learning and be left to humans.

IMPACT ON SOCIETY AND ENVIRONMENT

With the development of ever evolving AI-systems moving at a rapid pace, it is clear that it will be a key component in defining the future of humanity. There are great possibilities with the involvement of AI in our lives, as it has been proven so far in our technological advancements. AIs are already apparent in our lives as they provide us media recommendations, travel route recommendations, facial recognition and autonomous vehicles. However, there are also ethical dilemmas that need to be taken into consideration.

With the further development of AIs capable of autonomous decision making, it is unclear who will be responsible for its decisions and if we can trust that they are fair. The impact on society and environment also needs to be evaluated. In the medical field, the implementation of AI has the potential of saving many lives. The AI could for example be used for analyzing MRI-scans and finding cancer tumors more efficiently than a doctor is capable of. In a more advanced implementation, the AI could analyze a patient's symptoms and make a diagnosis that is a more accurate assessment than a doctor could ever make. It is a possibility that doctors will be fully replaced by an AI in the future. There are ethical dilemmas that arise in these cases. The most relevant one is arguably whether the AI bears any responsibility when making the wrong decisions. Decisions that could lead to fatal consequences. It is not certain if the hospital, the company that created the AI or the government making the rules and regulations that carries the responsibility for these unfortunate outcomes. From a legal standpoint and from a perspective of patient safety you are not able to put that responsibility on an AI, and therefore it always has to be a human responsible for approving treatment for a patient, preferably a medical professional. For these reasons, AI should not replace doctors. The AI could still perform very valuable tasks that could revolutionize the medical field, but should only have a role as a counselor, and a doctor will have to make the final decisions.

The technological advancement of machine learning will lead to more advanced AIs capable of increasingly complex tasks. An obvious consequence of this is that jobs that require repetitive movement are where AIs will be applied first as these are the easiest to learn. Factory workers and cashiers are examples of workers in risk of losing their jobs. A repercussion might occur from the people who have lost their income and governments across the world will be forced to act. A few handpicked jobs however might not be automated at all. These are jobs where the human connection and emotion is far more important than the economic gain. Such jobs are elementary/high school teachers, babysitters, kindergarten teachers, psychologists, care assistants etc. The majority of these jobs involve children and elderly people. Some may think that the economic gain is worth

the automation of these work areas, however, when asked if they would let their own children or parents be in care of AIs their answers would not be as positive.

Some believe that the automation of the majority of jobs will lead to mass unemployment. It is however, important to keep in mind that at the beginning of the industrial revolution, the majority of the population were farmers. They were faced with the same dilemma where the machines were robbing them of their jobs but as a result, new jobs were formed in factories. In the beginning of the revolution, the conditions for these workers were terrible, but over a long time great economic and social progress was achieved as a result. This is an example showcasing that replacing workers might give way for new jobs that will lead to further progress of humankind.

A high temporary unemployment rate is not desirable for anyone as it damages the population's mental health and sense of purpose. On the other hand, to hinder technological advancement should be seen as more unethical and is not, in our opinion, the correct decision to make as it would slow down mankind's economic and technological progress. These are part of the sole reasons for increased social progress. It is obvious that some regulations will have to be applied to AI but to completely halt its development is nothing short of madness.

Another consideration is that automated large-scale surveillance of workplaces and population now is feasible, due to the continued development of facial recognition algorithms. This, like most new developments, brings up many ethical problems. These include increased control over workers to increase their productivity indirectly making them slaves to their algorithm. We risk to trade our privacy, freedom of movement and control of our lives for a safer society, where mass surveillance is systematically used to disincentivize crime.

Of course, the answers to these problems are ambiguous as most ethical dilemmas are. However, the crux remains. It is hard to evaluate your privacy. With the greater usage of the internet, data is continuously generated by you and can be plugged into algorithms that analyze and track your behavior. This cannot be allowed to continue unchecked, therefore the introduction of privacy focused laws such as the General Data Protection Regulation (GDPR), while not perfect, are still a step in the right direction to curbing online monitoring and controlling of individuals. Already now, while the technology is still new, thousands of warehouse workers are constantly being ranked, evaluated and penalized by algorithms to increase their productivity, whilst creating a hostile work environment.

It is also important to discuss the environmental impact of AI since there can be both positive and negative impacts on the environment, depending on how the AI will be utilized. For example, the possibility to save resources can be enormous regarding many different fields, such as agriculture where smart decisions can be made on how to water as efficiently as possible, or how food should be distributed to minimize waste. But there is also a case where AI has a negative impact on the environment in how it is used in targeted ads that make us consume products we do not need. This is already implemented today, and will most likely increase in the future since the world economy is dependent on always making more money. The instances where companies care about the environment is when they can use it to promote their products, or if they get punished for not following environmental regulations. Therefore, it is important that society as a whole, continues developing environmental agreements and regulations.

Another aspect that affects the environment is that implementation of AI-systems requires a lot of computation power if you want to get decent results. Therefore there will be an increasing demand for both energy and computer components. For this reason, it will be important to use green energy to minimize the impact on the environment and enforce sustainable product development of components.

The possibilities for AI are almost endless and therefore it will be important for both lawmakers and regular people to remain observant on how the situation develops. The industrial revolution has taught us one thing: large technological advances bring great progress at the cost of venturing the rights for the general populace. For AI to develop sustainably, it will be essential that people stay alert and act to not get exploited in the future.

Playing Atari Breakout Using Deep Reinforcement Learning

Simon Jonsson and Jonas Lidman

Abstract– This report investigates the implementation of a *Deep Reinforcement Learning* (DRL) algorithm for complex tasks. The complex task chosen was the classic game Breakout, first introduced on the Atari 2600 console. The selected DRL algorithm was *Deep Q-Network* (DQN) since it is one of the first and most fundamental DRL algorithms. To test the DQN algorithm, it was first applied to CartPole which is a common control theory problem, using values describing the system as input. The implementation was then slightly modified to process images when employed for Breakout, in which it was successful. The application received a higher score than a professional human game tester. However, work remains to be done to achieve performance similar to state-of-the-art implementations of the DQN algorithm.

Sammanfattning– Denna rapport undersöker tillämpningen av en *Deep Reinforcement Learning* (DRL) algoritm för komplexa uppgifter. Den komplexa uppgift som valdes var Breakout från konsolen Atari 2600. DRL-algoritmen som användes var *Deep Q-Network* (DQN), eftersom det var en av de första och mest grundläggande DRL-algoritmer. För att kontrollera DQN-algoritmen tillämpades den först på CartPole, vilket är ett vanligt problem från reglerteknik, med tal som beskriver systemet som indata. Implementationen var sedan aningen modifierad för att kunna hantera bilder när den användes till Breakout, i vilken den presterade väl. Applikationen fick fler poäng än en professionell speltestare. Det finns dock andra implemeteringar som har fått högre poäng, och mer arbete behövs för att uppnå likvärdiga resultat.

Index Terms– Reinforcement learning, CartPole, Breakout, DQN

Supervisor: Damianos Tranos

TRITA number: TRITA-EECS-EX-2022:129

I. INTRODUCTION

How do we learn something without an explicit teacher? We try out a bunch of actions, see what happens and take note if what happened was good or bad. For example when an infant takes a random action such as waving its arms, and then sees and feels what happens, the infant will learn how its actions will affect the environment [1]. This coupling of action and response is relevant throughout our lives both when learning new tasks and in social environments.

Reinforcement Learning (RL) is the application of this paradigm in order to solve complex tasks. It does so by implementing an agent that tries out a large number of actions in an environment and tries to maximize the reward, see Fig. 1. For example if the environment is the computer game Snake, the agent has the controls up, down, left and right, and tries to maximize the game score. There are many different RL-algorithms that leverage this idea in different ways such as the TD algorithm (1988), Q-learning (1989) and the SARSA algorithm (1994) [1].

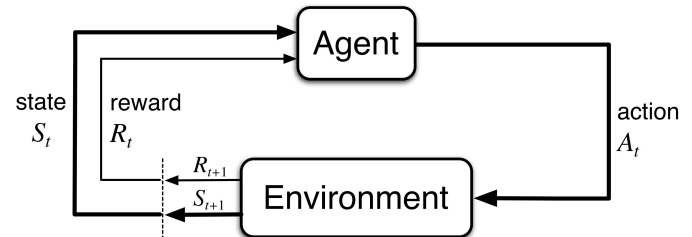


Fig. 1. Illustration of how the agent's action affects the environment and reward. [1]

One of the issues with previously mentioned RL-algorithms is that they do not scale well when the dimension of a problem grows, since the computational complexity scales exponentially. To solve this problem, we can use a *Neural Network* (NN) that approximates the environment, since NNs have been shown to be good universal learners. The combination of NNs and RL gives birth to *Deep Reinforcement Learning* (DRL) and creates new possibilities for solving more complex environments, than could be done with previous RL-algorithms [2].

A popular way to test DRL algorithms is to apply them to games, since they can be considered complex tasks, are easily available and are safe to use for experimentation. To do this efficiently an emulator can be used, such as the library openAI-gym which has a number of classic games, for example Breakout, and classic control theory problems such as balancing an inverted pole on a cart (CartPole) [3].

The purpose of the project is implementing a DRL-algorithm known as *Deep Q Network* (DQN), which is one of the first successful DRL-algorithms, on the Atari 2600 Breakout game. The objective of this work is to learn about the intricacies and testing of DQN.

II. THEORY

The fundamentals of Reinforcement Learning consist of a loop where the agent observes the state S_t and the reward R_t

of the environment, at the time t . With this information, the agent decides what action A_t it thinks is best. The environment then generates a new state S_{t+1} and reward R_{t+1} at timestep $t+1$ and is illustrated in Fig. 1. This loop continues until the state terminates, meaning the game is over. The chain of loops make up an *episode*.

A. Markov Decision Process

A game can be a *Markov decision process* (MDP), which means that only the current state and action affects the future of the game. To describe this more specifically, a few concepts will be introduced:

A *finite Markov decision process* means that the environment has a finite state spaces \mathcal{S} . There is also a set of finite actions \mathcal{A} that can be taken. Given a random state s and an action a , the probability for the next state and reward pairs can be described by equation (1).

$$p(s', r | s, a) = \Pr\{S_{t+1} = s', R_{t+1} = r | S_t = s, A_t = a\} \quad (1)$$

If equation (1) is fulfilled, that is that the probability for the next state only depends on the current state, the *Markov property* is satisfied. Given equation (1), one can compute the expected reward r for a given state action pair (s, a) :

$$r(s, a) = \mathbb{E}[R_{t+1} | S_t = s, A_t = a] \quad (2)$$

B. Policy

A policy π will be introduced and describes which action should be taken at a given state S . For every policy π we will associate the expected, discounted accumulative reward as:

$$V_\pi = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s \right] \quad (3)$$

where the discount factor $\gamma \in [0, 1)$ reflects the prioritization of short-term over long-term reward. The Q-value for a given policy describes the predicted value of taking a certain action in a specific state:

$$Q_\pi(s, a) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a \right] \quad (4)$$

From this, the optimal policy π_* is defined as the best possible policy, meaning that it maximizes the total reward. That is, if one is able to determine the optimal policy π_* , this is equivalent to one determining the best strategy for playing a game.

C. Q-learning

One way to find the optimal policy π_* is to learn the Q-values directly, which is accomplished by the Q-learning algorithm. The first step is to initialize the Q-values that describes the predicted value of all possible state and action pairs. This gives a Q-table that has the dimension $\mathcal{S} \times \mathcal{A}$. After that, an action is chosen according to an ϵ -greedy policy. This means that the algorithm takes a random action with a probability of ϵ and otherwise chooses the best known action

according to the Q-table. This is done to make sure that the agent explores all states. The final step is to update the Q-table which is done by using the *Bellman update*, described as follows:

$$Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$$

where $\alpha \in [0, 1]$ is the learning rate, which determines the size of an update step. $\max_a Q(S', a)$ describes the value of taking the action that gives the highest Q-value in the next state S' according to the current policy. To make the Q-table converge to the optimal policy π_* , all state and action pairs need to be continuously revised with the Bellman update, meaning that we need to visit all states and actions several times [1], resulting in Algorithm 1.

Algorithm 1 Q-learning from [1]

```

Initialize  $Q(s, a), \forall s \in \mathcal{S}, a \in \mathcal{A}$ , arbitrarily,
and  $Q(\text{terminal-state}, \cdot) = 0$ 
for each episode do
  Initialize  $S$ 
  repeat for each step of episode
    Given  $S$ , choose  $A$  using policy derived from
     $Q$  (e.g.  $\epsilon$ -greedy)
    Take action  $A$ , observe  $R, S'$ 
     $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$ 
     $S \leftarrow S'$ 
  until  $S$  is terminal

```

For small environments, this is a feasible method, but when the environments are large, such as video games, too many calculations are needed. Therefore we will present a deep reinforcement learning algorithm, known as *DQN*. To this end, we first precise the notion of a Neural Network.

D. Neural Network

A *Neural Network* (NN), is inspired by nature to try to mimic how a brain works. The brain consists of a number of biological neurons which are connected and trigger each other in chains, where different connections have different importance. This idea can be employed in computer science and Fig. 2 showcases how this would work for a single neuron. The inputs are either signals from other neurons multiplied with a signal specific weight, or information from the environment. A bias is also added, and is unique for every neuron. After that the neuron is activated using an activation function, in our case *Rectified Linear Unit* (ReLU) Fig. 3, which introduces the possibility of approximating non-linear problems. The neuron's output is then passed to other neurons, which work in the same way. When creating a NN, the nodes are combined into layers and make up *fully connected linear layers*. The neurons are only connected in one direction. If these layers are not part of the input or output of the NN, they are called hidden layers. An overview of this type of NN, can be viewed in Fig. 4.

Another type of layer is called *convolutional layer* and is commonly used in image processing. The reason for using

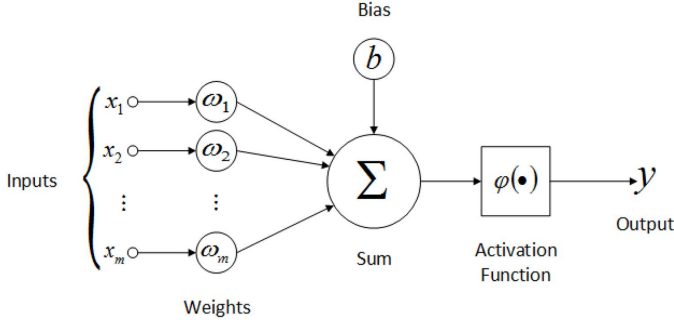


Fig. 2. Mathematical model of an artificial neuron. [4]

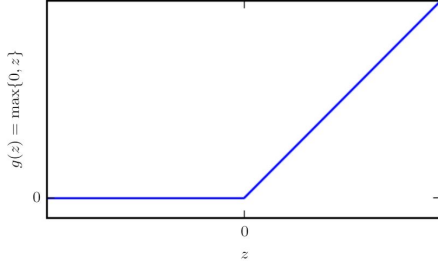


Fig. 3. Rectified Linear Unit (ReLU). [2]

convolutional layers when working with images instead of the previously introduced fully connected linear layers is to reduce the number of parameters by picking out the key features of an image [2]. This would reduce the computational cost, meaning that tasks can be solved more efficiently. Two of the convolutional layer parameters are the kernel size, which is the number of pixels processed together, and the stride, which is the number of pixels that the analysis window moves on each iteration. The convolution can include a number of kernels which each pick out different features. The convolution can be visualized in Fig. 5 where the kernel is the 3x3 shaded matrix on the 4x4 input channel. With a stride of 1, this produces a 2x2 output channel.

E. Deep Q-learning

Deep Q-Network (DQN) is an algorithm which uses a Neural Network that takes a state as input and estimates the Q-value for different actions. When playing the game, the agent will store experiences in a memory D. An experience at timestep t consists of the state S_t , reward R_t , action A_t and next state S_{t+1} . We will employ two NNs with the same architecture to stabilize learning: the action value-function Q and the target action-value function \hat{Q} that work in parallel. We denote the parameters (or weights) of Q and \hat{Q} as θ and $\hat{\theta}$, respectively. After a predetermined number of steps, the weights for the target \hat{Q} are set equal to those of Q . Similarly to Q-learning, ϵ -greedy is used to select an action. The main NN is then updated with the *Bellman update*, with the exception that the target network for Deep Q-learning is now used to estimate the expected Q-value in the next state:

$$Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a \hat{Q}(S', a) - Q(S, A)]$$

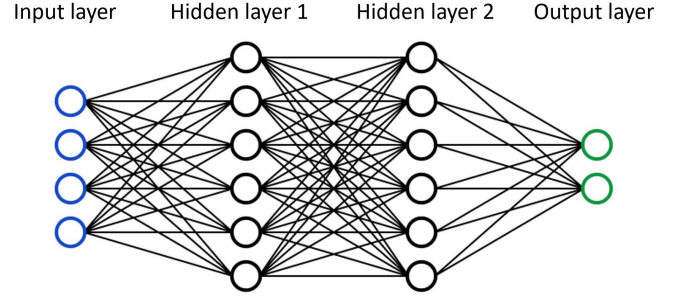


Fig. 4. Neural Network where the hidden layers are fully connected linear layers.

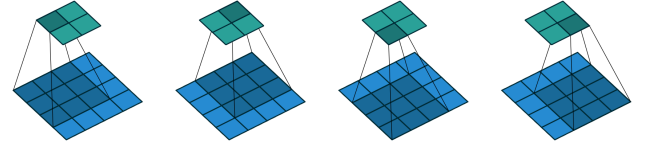


Fig. 5. Convolving a 3x3 kernel over a 4x4 input using a stride of 1. [5]

To update the NN, the first step is to calculate a loss and perform a step of stochastic gradient descent. This is done by sampling a small batch of experiences from the memory D randomly, to minimize the risk of using correlated data. The batch is then used to estimate maximal expected reward for the next state s' , using the target network \hat{Q} . This value is then compared with the Q-value from the current state s and the remembered action taken. The loss is calculated with the root mean squared method:

$$L = \mathbb{E}[r + \gamma \max_{a'} \hat{Q}(s', a') - Q(s, a)]^2 \quad (5)$$

and is used to perform a gradient descent step on the main networks weights θ with the ADAM optimizer [6]. The complete algorithm is described in Algorithm 2.

Initially, ϵ will be large so that many different strategies are explored. It then decays linearly (or exponentially) to reflect the algorithm's commitment to exploiting the best strategy it has learned.

III. CARTPOLE

An environment that is commonly used to test DRL algorithms is known as CartPole-v0 from the OpenAI library [3], and simulates an inverted pendulum, where the goal is to make the pole stand upright as long as possible, see Fig. 6.

A. Environment

The agent can control the cart in CartPole by exerting a force of 1 N on the cart either to the left or to the right, where the mass of the cart is 1 kg and the mass of the pole is 0.1 kg. The output from the CartPole environment is a vector, describing the state of the system at a given time: $S_t = [x_t, v_t, \theta_t, \omega_t]$, where x_t and v_t is the position respectively velocity of the cart, and θ_t and ω_t describe the angle and angular velocity of the pole. When an episode starts,

Algorithm 2 DQN with experience replay, from [7]

```

Initialize replay memory  $D$  with a capacity of  $N$ 
Initialize action-value function  $Q$  with random weights  $\theta$ 
Initialize target action-value function  $\hat{Q}$  with  $\hat{\theta} = \theta$ 
for every episode do
  Initialize starting state  $s_0$ 
  while Episode not terminated do
    sample  $p$  from  $U(0, 1)$ 
    if  $p > \epsilon$  then
      Select action  $a_t = \operatorname{argmax}_a Q(s_t, a_t)$ 
    else
      Select random action  $a_t$ 
    Execute action  $a_t$ 
    Observe next state  $s_{t+1}$  and reward  $r_t$ 
    Store  $(s_t, a_t, r_t, s_{t+1})$  in  $D$ 
  procedure (every  $K$  steps)
    Sample random batch  $(s_i, a_i, r_i, s_{i+1})$  from  $D$ 
    if Episode terminates at  $i+1$  then
       $y_i = r_i$ 
    else
       $y_i = r_i + \gamma \max_{a'} \hat{Q}(s_{i+1}, a')$ 
    Calculate  $(y_i - Q(s_i, a_i))^2$  and
    perform a gradient descent step on  $\theta$ 
  Update  $\theta = \hat{\theta}$  every  $C$  steps
  if Episode ends then
    set done to True
  else
    set  $s_t = s_{t+1}$ 

```

these values are sampled uniformly between ± 0.05 , and an episode ends if the moves too far to one side (± 2.4) or if the angle of the pole is too great ($\pm 12^\circ$). The agent receives one point each time step as long as the episode has not ended. If the score is greater than 200, the episode ends.

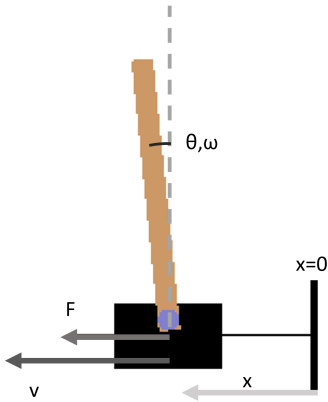


Fig. 6. Graphical depiction of the CartPole environment. Adapted from [3]

B. Evaluation

The CartPole environment is considered solved if the average score is greater than 195 over 100 consecutive episodes. To evaluate the algorithm's stability we executed the algorithm 100 times, with 2500 episodes each time, and collected the

score, average score and the average Q-value. To calculate the average Q-value, a large set of random states is passed through the NN which outputs their associated Q-values. If the average Q-value curve plateaus, it indicates that the agent does no longer improve. However, if it continues to increase, it means that the agent is still learning. Therefore it is useful to plot the average Q-value, since it gives an understanding of the agent's learning process. The average score is the average over the last 100 episodes.

C. Results

The hyper parameters for the CartPole evaluation are chosen according to TABLE II, with a NN-architecture described in TABLE I and with ϵ decreasing exponentially over 200 taken actions.

TABLE I
THE NEURAL NETWORK ARCHITECTURE USED FOR CARPOLE

Layer	Input	Output
Linear 1	4 nodes	32 nodes
Linear 2	32 nodes	2 nodes

TABLE II
MODEL PARAMETERS FOR CARPOLE

Hyperparameters	Value
Batch Size	32
Training frequency k	1
Target network update frequency C	2
Replay memory size N	500,000
Discount factor	0.99
Learning rate	0.0005
ϵ start	0.9
ϵ end	0.2

An average score at each episode for the 100 realisations is shown in Fig. 7. The average score reaches a maximum of around 188 points, which is insufficient for solving the CartPole environment. However, all executions reach the solved condition of 195 points at some point which is apparent in Fig. 9 where three of the realisations are plotted. The average scores tend to oscillate and are not at a constant score. According to Fig. 8 the average Q-value plateaus at around 1,500 episodes, indicating that the model can not be improved further with training.

D. Discussion

Fig. 7 shows some interesting characteristics of the DQN algorithm in that the average does not reach 195 over 100 executions, indicating that the DQN algorithm is slightly unstable and can deviate from a good solution.

In Fig. 9, three random realisations are plotted, showcasing the oscillation and that all of the three executions do in fact reach the target of 195 at some point. Since the environment resets after reaching 200 points, the average is sensitive to a low score run and it can not be compensated with a high score run. Therefore, the model needs to be very stable to maintain a constant average over multiple episodes.

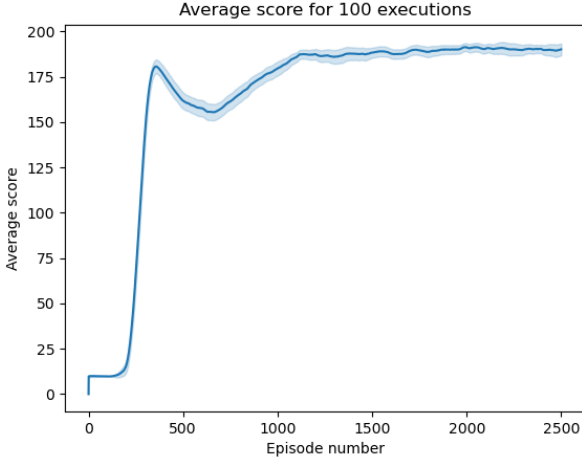


Fig. 7. The blue line shows the average score of 100 executions of the algorithm. The shaded area indicates the confidence interval of 95% for the blue line.

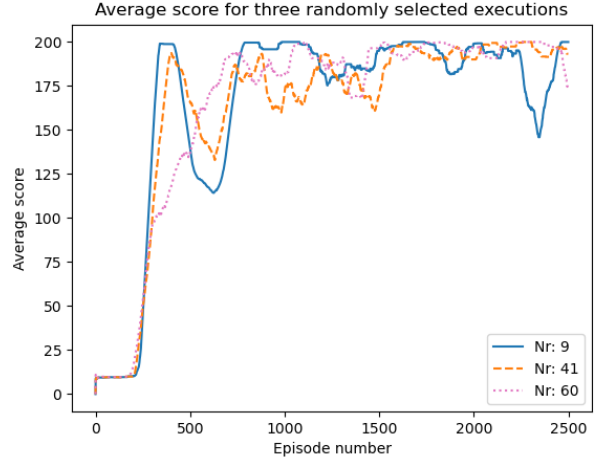


Fig. 9. The three plots are randomly selected executions of the algorithm, showcasing the oscillation of the average score.

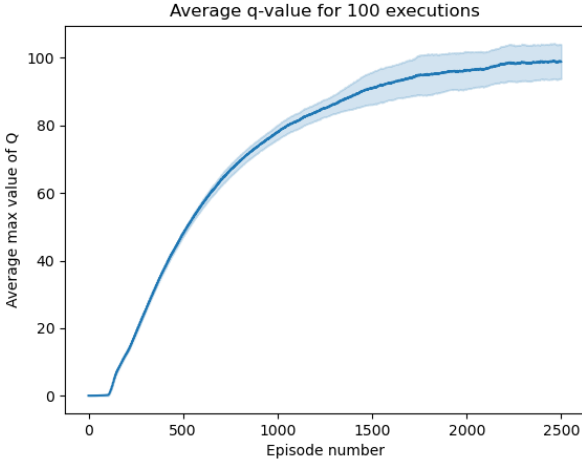


Fig. 8. The blue line shows the average Q-value of 100 executions of the algorithm. The shaded area indicates the confidence interval of 95% for the blue line.

In Fig. 8 the variance increases after 1,000 episodes, which implies that there is a lot of deviation from the mean. The reason for this is that stochastic gradient descent requires uncorrelated data to converge as well as the target for the loss function being fixed, neither of which are fulfilled. This is a general problem with RL since we have temporarily correlated data and target values that are changing, due to them being estimated with a NN, which itself is updated at a certain frequency. Experience replay and the target network attempt to address these issues, but they do not do so perfectly, leading to the lingering instability (and hence, variance) of the algorithm.

IV. BREAKOUT

As a more complex environment to test our DQN-implementation, we have chosen the classical video game Breakout for Atari 2600.

A. Environment

The purpose of the game is to bounce a ball on a pad to hit bricks. When a brick is hit, it gives the player one or several points and the brick disappears. Bricks further up give more points, but also make the ball move faster. OpenAI [3] provides the environment as *BreakoutNoFrameskip-v4* which generates an image such as Fig. 10, and tells the agent if it got a reward for hitting a brick. The following actions are available: 'NOOP', where nothing is done, 'FIRE', which is needed to start a round by releasing the ball, 'RIGHT', which moves the paddle to the right and 'LEFT' which moves the paddle to the left. The player also has 5 lives, and loses a life if it misses the ball.

However, to make the task easier to complete, meaning less computations are needed, we have made a few alterations to the environment, that translate into how a human would view the game. The first one is that every time the player starts a new game or round, the 'FIRE' action is taken automatically dropping the ball immediately. This is simply to remove the time where the game is not played. The second alteration is in regard of losing a life. From a human stand point it is natural that losing a life is bad, but *BreakoutNoFrameskip-v4* does not give any feedback when losing a life. To help train the neural network, we make the agent believe the game is over if it loses a life, meaning that it believes it can not get any more points. A third change is that only every fourth frame is generated, since frames directly after each other provide very similar information.

B. Preprocessing

To make the Neural Network handle the problem easier a few procedures are performed to the image from the *BreakoutNoFrameskip-v4* environment Fig. 10. First of all, colours are removed, since they do not provide any additional information Fig. 11. Secondly, the score and life counters as well as part of the frame is cropped out, leaving only the actual play area. This means that the image goes from 210x160

pixels to 160x144 pixels Fig. 12. The agent does not need know or learn how many lives it has or what the score is (the reward is provided separately from the image as stated in the environment description). The image is then compressed to a 84x84 sized image Fig. 13 on which a binary filter is applied to producing an image with only completely black or completely white pixels Fig. 14. Four of these frames are stacked to conserve perceived velocity in the system, so as to fulfill the Markov property, and becomes the input to the Neural Network.

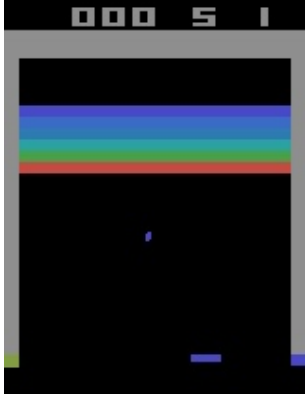


Fig. 10. The original image produced by the environment, 210x160 pixels.

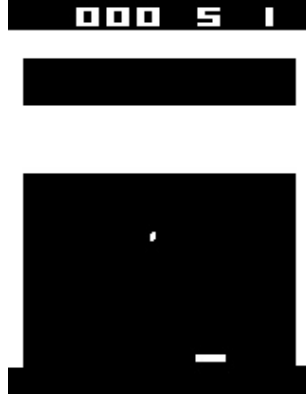


Fig. 11. Black and white filter, 210x160 pixels.

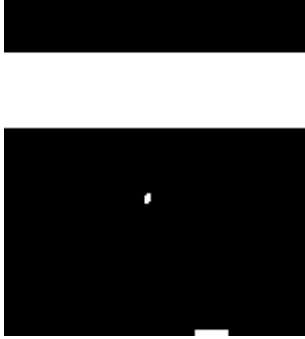


Fig. 12. Cropping so only the play area is visible, 160x144 pixels.

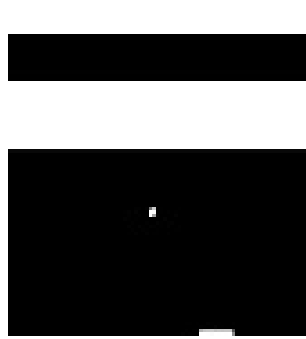


Fig. 13. Compressing to 84x84 pixels, which introduces some grey pixels.



Fig. 14. Binary filter making every pixel black or white, which is the input to the Neural Network, still 84x84 pixels.

C. Neural Network

To estimate the Q-value function a neural network was created. The network consists of five layers, see Fig 15.

The first three layers are convolutional, which, as mentioned earlier, reduce the computational cost when the input consist of images. The two remaining are linear layers. For the first four layers a ReLU is used as an activation function. The parameters of the layers are shown in TABLE III.

TABLE III
THE NEURAL NETWORK ARCHITECTURE USED FOR BREAKOUT

Layer	Input	Output	Kernel size	Stride
Conv 1	4 channels	32 channels	8	4
Conv 2	32 channels	64 channels	4	2
Conv 3	64 channels	64 channels	3	1
Linear 1	7-7-64 nodes	512 nodes		
Linear 2	512 nodes	4 nodes		

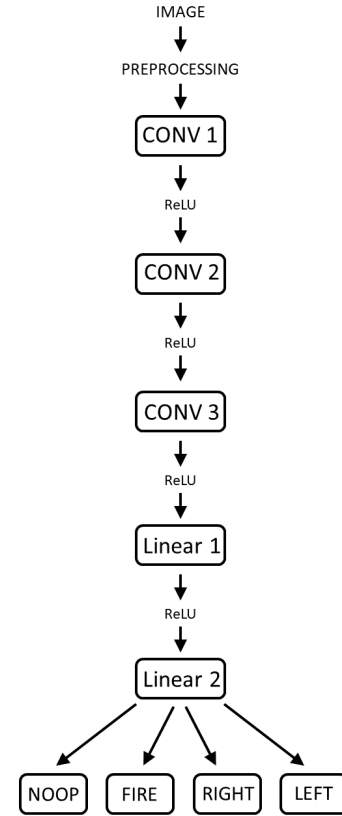


Fig. 15. The Neural Network architecture used for Breakout. The NN consists of 5 layers of which the first three are convolutional layers and the last two are linear. The NN architecture has been taken from [8] which is the GitHub of the implementation of the DeepMind's Nature article [7].

D. Evaluation

The solution will be evaluated in three ways: firstly, the average score and estimated Q-value when the agent is training where random actions will be taken with a probability of ϵ . The final trained agent will be evaluated with the 'FIRE' button, being pressed, dropping the ball automatically, since this is how the agent was trained. DeepMind's DQN algorithm [7] was evaluated by letting the agent play 30 games of Breakout, with an ϵ of 0.05. We will do the same, to be able to compare our agent's performance, but it would be more reasonable to

use $\epsilon = 0$ so the agent makes all decisions. It is important to keep in mind that we have simplified the environment. Another interesting comparison is with the score of a professional human game tester, according to DeepMind’s paper [7].

E. Results

The agent was trained for 200 epochs, where one epoch represents 50,000 learning batches. The hyper parameters are displayed in TABLE V, where ϵ decreased linearly over 100,000 actions. Two plots were generated to keep track of the training process. The first one is the average score per episode plotted against the number of training epochs, see Fig. 16. It should be noted that one episode for the agent during training is only one life, whereas in the final evaluation on episode consist of five lives. Fig. 17 shows the maximum average Q-value for 1000 randomly generated states. The final model is evaluated by letting the agent play for 30 episodes, see Fig. 18. TABLE IV shows the final score for our trained model, as well as the score of DeepMind’s DQN, a random agent and two other algorithms.

TABLE IV
BREAKOUT SCORES FOR DIFFERENT AGENTS FROM [7], AND C2A’S IMPLEMENTATION

Random Play	Best Linear Learner	Contingency (SARSA)	Human	DQN DeepMind 200 epochs	DQN C2A 200 epochs
1.7	5.2	6.1	31.8	401.2 (± 26.9)	215 (± 131.6)

TABLE V
HYPERPARAMETERS FOR BREAKOUT

Hyperparameters	Value
Frames stacked	4
Batch size	32
Training frequency k	4
Target network update frequency	10,000
Replay memory size N	10,000
Discount factor	0.99
Learning rate	0.00001
ϵ start	1
ϵ end	0.05

F. Discussion

When implementing the DQN algorithm, we achieved an average score of 215 (± 131.6) TABLE IV. Comparing this score to that of random play, 1.7, and other reinforcement learning algorithms (Best linear learner (5.2) and SARSA (6.1)) the DQN algorithm performs significantly better. This implies that the DQN strategy was implemented correctly and is a good solution for complex environments. However, our implementation with 200 epochs did not achieve the same score as DeepMind’s DQN score 401.2 (± 26.9) with 200 epochs. We also have a significantly greater standard deviation when evaluated in the same way (30 episodes with $\epsilon = 0.05$), meaning that our agent is not as stable as DeepMind’s agent.

The most likely reason that our agent does not perform as well as DeepMind’s, is simply because our implementations are not the same. We have chosen many of the hyperparameters, and the Neural Network architecture, to be the same as DeepMind’s in their paper [7], but we use a different

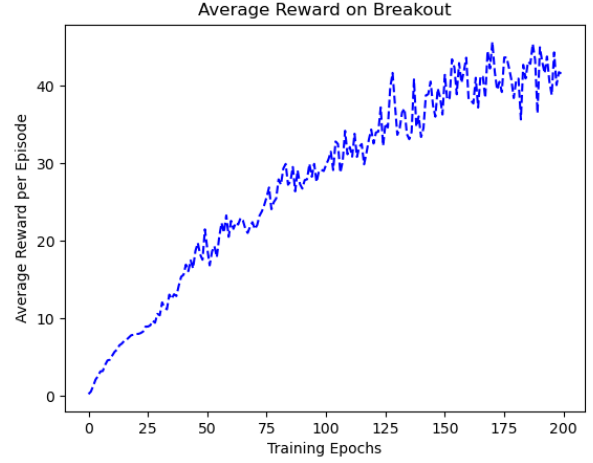


Fig. 16. The average score achieved per episode in Breakout during training. One epoch lasts for 50,000 learning batches. An episode ends after the agent has lost one life.

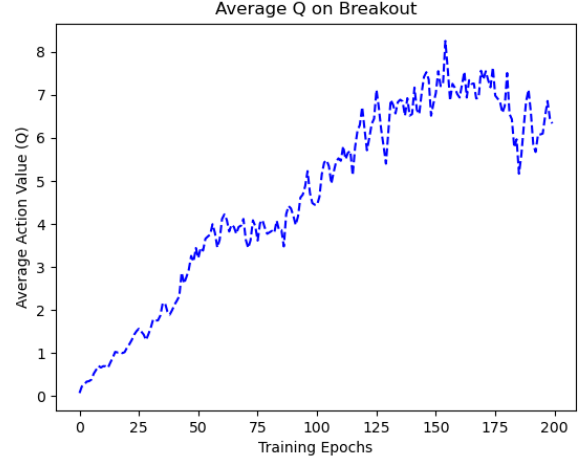


Fig. 17. The average action value (Q) achieved per episode in Breakout during training. One epoch lasts for 50,000 learning batches. An episode ends after the agent has lost one life.

optimizer. Moreover, our current hardware limits us when it comes to replay memory size, and we can not have a size of 1,000,000, but are restricted to a memory of an order of magnitude less in size. It should also be stated that we did not use the same emulator for Breakout. Furthermore, DeepMind [7] do not state how the score was achieved in the paper: if it was over several realisations or if they manually selected the one that performed the best. To achieve better results, other hyperparameters could be investigated.

It is also interesting to note that the average Q-value stops increasing toward the end of the realisation, which can be viewed in Fig. 17, indicating that this combination of hyper parameters and model architecture is not learning any more. However, the average reward is still slightly increasing at the end of the realisation, meaning that the agent is still learning.

Another important aspect to bring up is that we altered the environment to make it easier to train by making the

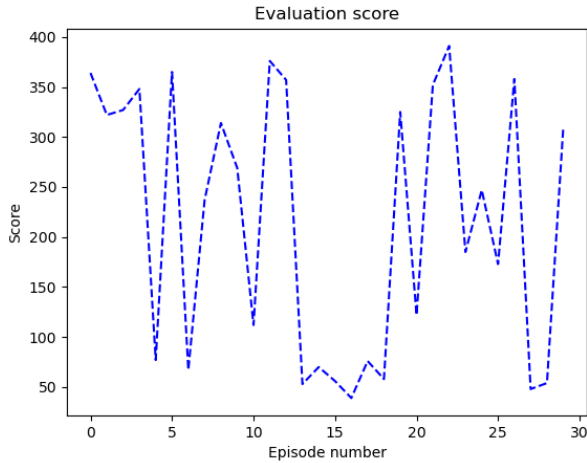


Fig. 18. The score of the trained agent playing for 30 episodes, with an average of 215 and a standard deviation of 131.6. For the evaluation an episode is finished after the agent has lost 5 lives.

environment press the 'FIRE' button in the beginning of each round. In DeepMind's implementation they solved this in a slightly different manner by resetting the environment if the 'NOOP' action was chosen 30 times in a row. We also cropped the screen to remove the scores, which DeepMind did not do.

V. CONCLUSION

The Deep Reinforcement Learning algorithm Deep Q-Network was successfully implemented on the Atari Breakout game using images as information, reaching performance better than that of a human game tester, but performs worse than that of the state-of-the-art applications.

CODE REPOSITORY

For the full code repository, please view: <https://github.com/JonasLidman/DQN-for-Bachelor-Thesis-EECS-KTH-2022>

ACKNOWLEDGEMENT

We, the authors, would like to thank our supervisor Damiános Tranos for the guidance and support he has provided to make this project possible

REFERENCES

- [1] R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*, ser. Adaptive computation and machine learning. Cambridge, Massachusetts London, England: The MIT Press, 2018.
- [2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. Cambridge, Massachusetts London, England: MIT press, 2016.
- [3] M. LLC. (2022, Feb.) A toolkit for developing and comparing reinforcement learning algorithms. [Online]. Available: <https://gym.openai.com/>
- [4] R. M. S. d. Oliveira, R. C. F. Araújo, F. J. B. Barros, A. P. Segundo, R. F. Zampolo, W. Fonseca, V. Dmitriev, and F. S. Brasil, "A system based on artificial neural networks for automatic classification of hydro-generator stator windings partial discharges," *Journal of Microwaves, Optoelectronics and Electromagnetic Applications*, vol. 16, no. 3, pp. 628–645, 2017.
- [5] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," 2016.

- [6] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980v9*, 2015.
- [7] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [8] DeepMind. (2022, May) A toolkit for developing and comparing reinforcement learning algorithms. [Online]. Available: <https://github.com/deepmind/dqn>

Deep Reinforcement Learning for Card Games

Rayan Cali and Oscar Tegnér Mohringe

Abstract—This project aims to investigate how Reinforcement Learning (RL) techniques can be applied to the card game Limit Texas Hold'em. RL is a type of machine learning that can learn to optimally solve problems that can be formulated according to a Markov Decision Process.

We considered two different RL algorithms, Deep Q-Learning (DQN) for its popularity within the RL community and Deep Monte-Carlo (DMC) for its success in other card games. With the goal of investigating how different parameters affect their performance and if possible achieve human performance.

To achieve this, a subset of the parameters used by these methods were varied and their impact on the overall learning performance was investigated. With both DQN and DMC we were able to isolate parameters that had a significant impact on the performance.

While both methods failed to reach human performance, both showed obvious signs of learning. The DQN algorithm's biggest flaw was that it tended to fall into simplified strategies where it would stick to using only one action. The pitfall for DMC was the fact that the algorithm has a high variance and therefore needs a lot of samples to train. However, despite this fallacy, the algorithm has seemingly developed a primitive strategy. We believe that with some modifications to the methods, better results could be achieved.

Sammanfattning—Detta projekt strävar efter att undersöka hur olika Förstärkningsinlärning (RL) tekniker kan implementeras för kortspelet Limit Texas Hold'em. RL är en typ av maskininlärning som kan lära sig att optimalt lösa problem som kan formuleras enligt en markovbeslutsprocess.

Vi betraktade två olika algoritmer, Deep Q-Learning (DQN) som valdes för sin popularitet och Deep Monte-Carlo (DMC) valdes för dess tidigare framgång i andra kortspel. Med målet att undersöka hur olika parametrar påverkar inlärningsprocessen och om möjligt uppnå mänsklig prestanda.

För att uppnå detta så valdes en delmängd av de parametrar som används av dessa metoder. Dessa ändrades successivt för att sedan mäta dess påverkan på den övergripande inlärningsprestandan. Med både DQN och DMC så lyckades vi isolera parametrar som hade en signifikant påverkan på prestandan.

Trots att båda metoderna misslyckades med att uppnå mänsklig prestanda så visade båda tecken på upplärning. Det största problemet med DQN var att metoden tenderade att fastna i enkla strategier där den enbart valde ett drag. För DMC så låg problemet i att metoden har en hög varians vilket innebär att metoden behöver mycket tid för att tränas upp. Dock så lyckades ändå metoden utveckla en primitiv strategi. Vi tror att båda dessa metoder med ett par modifikationer skulle kunna uppnå ett bättre resultat.

Index Terms—Reinforcement Learning, Deep Q-Learning, Deep Monte-Carlo, Poker.

Supervisors: Alessio Russo

TRITA number: TRITA-EECS-EX-2022:130

I. INTRODUCTION

Artificial intelligence (AI) is a hot topic of research, and, recently an area called Machine Learning (ML) has started to

gain a lot of traction with increasing amounts of research being done [1]. Within ML, a theory called Reinforcement Learning (RL) has been developed, which aims to allow algorithms to learn more like humans by using trial and error. Simply this means that the algorithm interacts with a given system and generates data. Then this data is used to train the model as stipulated in [2].

Games provide a good medium to test out the efficiency and scalability of these algorithms. In [3], a team trained an RL agent to play the game "DotA 2". This AI would eventually go on to beat the reigning world champions, becoming the first AI system to do so in an e-sport. When compared to other groundbreaking achievements in chess [4] and GO [5], "DotA 2" has a continuous state and action space and it is only partially observable, i.e. you are unable to see what your opponent is doing.

In this report two RL algorithms are considered and implemented for Limit Texas Hold'em, which is a popular version of poker. While not as complex as "DotA 2" with continuous action and state spaces it still offers challenges with partial observability, planning a strategy, a rather large state space and a variable discrete action space.

The main goal of the project was to observe how different parameters affected the RL agents ability to learn. In order to do this, two different RL methods were used, namely Deep Q-learning and Deep Monte-Carlo. With the latter only being implemented for Limit Texas Hold'em.

A rundown of Markov decision processes, RL, Q-learning, Deep Q-learning and Deep Monte-Carlo are given in the theory section. The implementation and results of Deep Q-learning implemented for Cartpole are given in the Cartpole section. In the following two sections the implementation of Deep Q-learning and Deep Monte-Carlo for Limit Texas Hold'em are discussed. Lastly, we provide some insights and ideas for improvements.

II. THEORY

A. Markov Decision Process

A finite Markov decision process (MDP) is a stochastic process which can be used to describe systems and can be characterized as a tuple (S, A, p, r, γ) . S is the state space for the system, likewise A is the action space for the MDP. $p(s'|s, a)$ is the probability of transitioning to a state $s' \in S$ given a state s and an action a . $r(s, a)$ is simply the reward function for a given state-action pair (s, a) . Lastly, $\gamma \in (0, 1)$ is the discount factor which determines how much future rewards are discounted; a more detailed explanation can be found in [2].

The general goal in MDPs, is to compute a policy π^* that maximizes the total collected reward. This is called the optimal

policy. Policies are mapping from states to action $\pi : S \rightarrow A$, these can be deterministic or stochastic.

B. Reinforcement Learning

In reinforcement learning (RL) the dynamics of the selected system is not fully known, i.e. the reward and transition functions are unknown. Thus the main goal of RL is to find ways of computing an optimal policy by interaction with a given MDP. An RL agent is an implementation of an algorithm that interacts with an MDP, and learns from its observations.

There are many different techniques that can be used to compute the optimal policy. However, information about the reward is required, thus we define R_t as the collected reward at time t given a state-action pair (s_t, a_t) . Then it is possible to define the total discounted reward:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad (1)$$

Using 1 it is possible to define the value-function and the state-action function, which will be referred to as the Q -function, following a policy π .

$$\begin{aligned} V^\pi(s) &= \mathbb{E}[G_t | S_t = s, \pi] \\ Q^\pi(s, a) &= \mathbb{E}[G_t | S_t = s, A_t = a, \pi] \end{aligned} \quad (2)$$

It is possible to rewrite both the value-function and Q -function in a recursive form.

$$\begin{aligned} V^\pi(s) &= \sum_a \pi(s, a) \left[r(s, a) + \gamma \sum_{s'} p(s' | a, s) V^\pi(s') \right] \\ Q^\pi(s, a) &= r(s, a) + \gamma \sum_{s'} p(s' | a, s) \sum_{a' \in A} \pi(a' | s') Q^\pi(s', a') \end{aligned} \quad (3)$$

The recursive form found in 3 is called the Bellman expectation equations. If given an MDP, the optimal value-function and Q -function obey the expectation equations found in 4.

$$\begin{aligned} V^*(s) &= \max_a \left[r(s, a) + \gamma \sum_{s'} p(s' | a, s) V^*(s') \right] \\ Q^*(s, a) &= r(s, a) + \gamma \sum_{s'} p(s' | a, s) \max_{a' \in A} Q^*(s', a') \end{aligned} \quad (4)$$

The value function of a certain policy π is simply the function $V : S \rightarrow R$ that returns the discounted total reward given an initial state s , following the policy π . The Q -function represents the value of taking an action a in state s , and then following the policy π , $Q : S \times A \rightarrow [0, 1]$.

C. Q-Learning

To explain what Deep Q-Learning (DQN) is, Q-learning has to first be explained. Q-learning is an algorithm that constantly updates the Q -values after every iteration.

$$Q_t(s_t, a_t) = Q_{t-1}(s_t, a_t) + \alpha_t (y_t - Q_{t-1}(s_t, a_t)) \quad (5)$$

Here α_t is a sequence of values that need to satisfy the Robbins Monroe conditions [6]. The conditions are that the sum of all α_t have to be greater than infinity while the sum of the squares of α_t should be less than infinity. In the rest of the report α_t will be called the learning rate.

$$y_t = R_t + \gamma \max_a Q_t(s_t, a) \quad (6)$$

What this algorithm then does is converging to the optimal Q -function, in other words, it finds the optimal action to receive the maximum reward for a given state [6].

There is however one important requirement for convergence. Each state-action pair must be visited infinitely often. This is of course not possible if the policy is only greedy. Therefore a behavior policy is introduced that is epsilon greedy. This behavior policy has a probability of ϵ to pick an action uniformly at random and a probability of $1-\epsilon$ to instead pick a greedy action which essentially means that it picks the action that maximizes the action value (reward). This ensures exploration for the agent so that all state-action pairs are visited which leads the agent to find the optimal value function Q^* which in turn finds the optimal policy π^* .

D. Deep Q-Learning

Using Q-learning however can be problematic when the state-space is too large as all state-action pairs have to be updated after every iteration. This is where Deep Q-learning is used instead. The "deep" part refers to using a neural network as a function approximator of the value function. Neural networks attempt to mimic the human brain's ability to recognize patterns [7]. They contain layers of nodes and algorithms and by feeding it information, for example a picture of a cat, can determine if the picture does in fact contain a cat by outputting a number between 0 and 1 where 1 would be a 100 percent chance of the picture containing a cat. Instead of updating the state-action pairs, it instead updates the weights of the network through a loss function, so it can converge towards the target. Weights are essentially a measurement of how strong a connection two neurons have in a neural network [8].

$$L(\theta') = \mathbb{E}[(y - Q_{\theta'}(s, a))^2] \quad (7)$$

Q is the approximated value function with weights θ and y is the target defined in equation (6). To minimize L so that the value network reaches the target network, the gradient of the loss function is calculated. With the help of the gradient the weights are updated so that the loss function becomes smaller.

$$\Delta \theta = -\alpha \nabla L(\theta') \quad (8)$$

This method is called stochastic gradient descent (SGD).

It is easy to notice now that a problem arises. The target uses the definition of Q itself which means that when the weights of Q are updated, so is the target. In other words, the function approximator never reaches the target. To solve this, we define a target function which is essentially another neural network. After every N amount of episodes, the target network copies the weights of Q and then stays set. This way the SGD

manages to converge and find the optimal weights before the target network gets updated again.

For SGD to work and have a chance at converging, another issue has to be solved. Taking the expected value in 8 is problematic as it has to be done for all state-action pairs which was what we were trying to avoid. A solution for this is to sample some Q -values and calculate the gradient of the loss that way instead. The sampling has to be completely random to avoid bias. By saving the state, action and reward in every time step t in a replay-memory, a batch of the replay-memory can be sampled and used for calculation of the gradient (SGD) at every step. The Deep Q-learning algorithm can be seen in algorithm 1.

Algorithm 1 The DQN algorithm

```

Initialize  $\alpha, \gamma$ 
Initialize Policy net  $Q$  and Target net  $Q'$ 
Initialize Replay memory  $D$  with size  $N$ 
Initialize  $\epsilon, \epsilon_{min}$  and  $\epsilon_{decaysteps}$ 
for  $n = 0$  to Number of episodes do
  Reset the environment
  for  $t = 1$  to  $T$  do
    Observe state  $s_t$ 
    if  $\epsilon > \epsilon_{min}$  then
      Decay  $\epsilon$  with  $\frac{1}{\epsilon_{decaysteps}}$ 
    end if
     $a_t \leftarrow \begin{cases} \operatorname{argmax}_a Q(s_t, a) & \text{with probability } (1-\epsilon) \\ \text{random action} & \text{with probability } \epsilon \end{cases}$ 
    Observe reward  $r_t$ , state  $s_{t+1}$  and done
    Store transition  $(s_t, a_t, r_t, s_{t+1}, done)$  in  $D$ 
    if  $D$  sufficiently filled then
      Sample a batch of transitions  $(s_k, a_k, r_k, s_{k+1}, done_k)$ 
      Perform a gradient step on the policy net  $Q$  with learning rate  $\alpha$ 
    end if
    Every  $M$  steps update  $Q' \leftarrow Q$ 
    if done then
      Terminate episode
    end if
  end for
end for

```

E. Deep Monte-Carlo

Deep Monte-Carlo (DMC) much like DQN relies on approximating the Q -value of actions using a deep neural network. What sets these methods apart is the fact that Monte-Carlo (MC) methods are only applicable to episodic tasks. Which means that it only can be applied to certain systems.

This is because DMC is unlike DQN which relies on bootstrapping (updating the Q -values while an episode is in progress), MC methods approximate the Q -value directly from the trajectory of an episode. Since MC cannot bootstrap, replay memories cannot be used and as such MC is generally less data efficient and has a higher variance than DQN since each generated episode can only be used for training once. The DMC algorithm used in this project can be found in algorithm 2.

Algorithm 2 The DMC algorithm

```

Initialize  $\alpha, \gamma$ 
Initialize the net  $Q$ 
Initialize  $\epsilon, \epsilon_{min}$  and  $\epsilon_{decaysteps}$ 
for  $n = 0$  to Number of episodes do
  Reset the environment
  Initialize or clear trajectory list  $T$ 
  for  $t = 1$  to  $T$  do
    Observe state  $s_t$ 
    if  $\epsilon > \epsilon_{min}$  then
      Decay  $\epsilon$  with  $\frac{1}{\epsilon_{decaysteps}}$ 
    end if
     $a_t \leftarrow \begin{cases} \operatorname{argmax}_a Q(s_t, a) & \text{with probability } (1-\epsilon) \\ \text{random action} & \text{with probability } \epsilon \end{cases}$ 
    Observe reward  $r_t$ , state  $s_{t+1}$  and done
    Store transition  $(s_t, a_t, r_t)$  in  $T$ 
    if done then
      Propagate the reward through list  $T$ 
      for  $k = t - 1$  to  $1$  do
         $r_k \leftarrow r_k + \gamma r_{k+1}$ 
      end for
      Perform a gradient step on the net  $Q$  with learning rate  $\alpha$ 
      Terminate episode
    end if
  end for
end for

```

III. CARTPOLE

In this section a brief explanation of the Cartpole environment is provided followed by a short discussion of the results. The main goal of this section was to confirm that the DQN implementation was working as intended.

A. Environment

The Cartpole environment consists of an inverted pendulum attached to a car, the main task is to move the cart to the right or to the left in order to balance the pole as seen in figure 1. In this environment the state space S is a continuous subset of \mathbb{R}^4 and the action space A is discrete with two different actions.

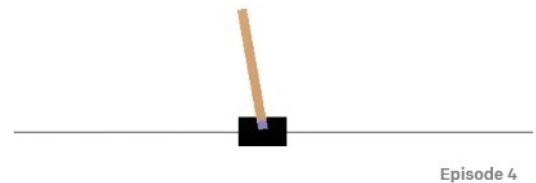


Fig. 1. A screenshot of the rendered Cartpole environment.

The input to the network is an array that contains the current positions and velocities of the cart and pole. Then the network takes an action. If the pole stays upright the network gets a reward of 1, or if it falls to the ground the episode ends with a reward of 0.

If the network manages to keep the pole upright for 200 steps the environment terminates, thus the maximum obtainable reward is 200. In this project we considered the environment solved if the DQN algorithm achieved a reward of 190 or greater for 5 episodes in a row on average of 10 independent repetitions.

B. Results & Discussion

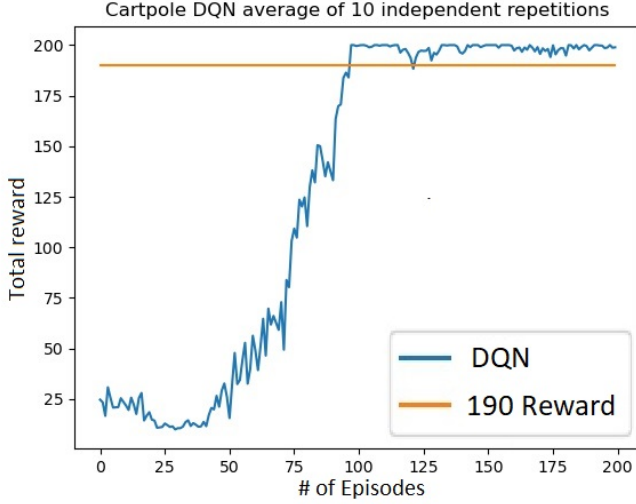


Fig. 2. Cumulative the reward for 10 independent repetitions.

The environment was solved according to the standards that were defined in the previous section, the results for this can be seen in figure 2. The parameters used in this run can be seen in table I.

TABLE I
PARAMETERS FOR CARPOLE

Parameter	Value
Number of neurons	256
Discount factor	0.99
Learning rate	10^{-2}
ϵ	0.9
ϵ_{min}	0.05
$\epsilon_{decaysteps}$	158 steps
Batch size	64
Target update	10
Memory size	1000000
Number of episodes	200

As expected it is quite easy to implement a DQN algorithm that solves Cartpole. Since the focus of the project was applying RL to card games, Cartpole was more a proof of concept than anything else and served as a demonstration that our implementation of DQN was working as expected.

IV. DQN & LIMIT TEXAS HOLD'EM

In this section an explanation of the Limit Texas Hold'em environment is provided. Followed by the results of our DQN implementation when interacting with the environment.

A. Environment

The environment used to simulate Limit Texas Hold'em is provided by the repository RL Card in [9]. In the environment the state is represented as a 72 element long list of booleans, the encoding used can be found in table II.

Table III shows the action space A and its encoding for the environment. In Texas Hold'em every action is not always available, this is called a variable action space. In practice this means that depending on the state certain actions are illegal. For example, you cannot "call" if your opponent has not raised in their turn.

The reward in the environment is milli big blind per hand (mbb/h). For example an agent that always folds is going to attain a reward of $-750 mbb/h$. Since the big blind is a bet of 1 and a small blind is a bet of 0.5. The reward increases each time a player picks "raise", depending on the outcome of the episode the agent gets reward for winning or punishment for losing equal to the won or lost bet.

TABLE II
STATE ENCODING IN RL CARD LIMIT TEXAS HOLD'EM

Index	Meaning
0-12	Spade A - Spade K
13-25	Heart A - Heart K
26-38	Diamond A - Diamond K
39-51	Club A - Club K
52-56	Raise in round 1
57-61	Raise in round 2
62-66	Raise in round 3
67-71	Raise in round 4

TABLE III
ACTION ENCODING IN RL CARD LIMIT TEXAS HOLD'EM

Action	Action ID
Call	0
Raise	1
Fold	2
Check	3

B. DQN vs Rulebased Agent

The evaluation of the DQN algorithm was done using two methods. The first method was to let the DQN agent learn by facing a rule based agent provided by RL card for 13 000 episodes. To measure the performance, a graph of the reward was created after every iteration of 13 000 episodes, as seen in figure 3. This was done a total of 16 times, however, some parameters were changed after each iteration. The parameter-combination for each plot can be seen in table IV. In the table, learning rate has been shortened "LR", replay memory size to "RMS", discount factor to "DF" and number of neurons to "Neurons".

Something very obvious with the result is that a lower discount factor leads to a worse DQN agent. This is because the only difference between the top two graphs and the dashed

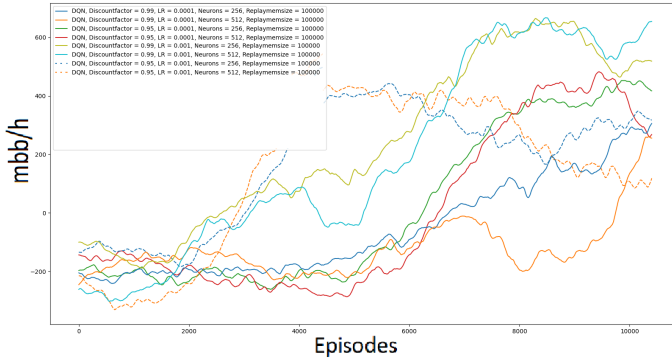


Fig. 3. *mbb/h* for the DQN agent. Positive value means that the DQN agent is winning. See figure 4 for a zoomed in view. The legend can be found in table IV.

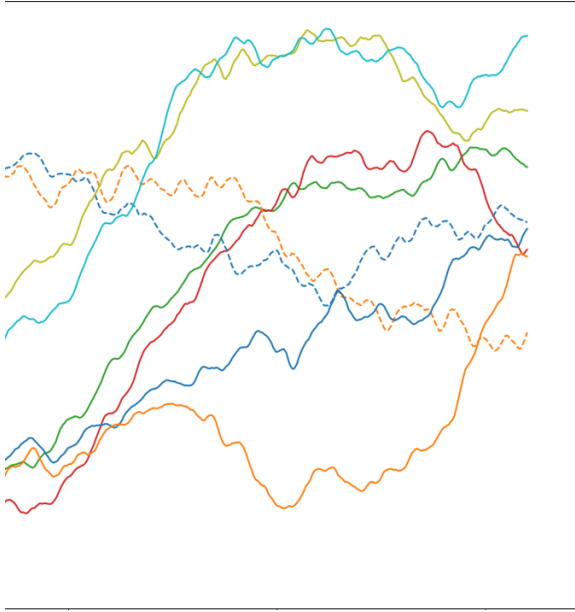


Fig. 4. Zoomed in view of figure 3.

TABLE IV
PARAMETERS FOR DQN AGENT

color	LR	RMS	DF	Neurons
dark blue	10^{-4}	10^5	0.99	256
light green	10^{-3}	10^5	0.99	256
orange	10^{-4}	10^5	0.99	512
dark green	10^{-4}	10^5	0.95	256
red	10^{-4}	10^5	0.95	512
light blue	10^{-3}	10^5	0.99	512
dashed blue	10^{-3}	10^5	0.95	256
dashed orange	10^{-3}	10^5	0.95	512

graphs in figure 4 is a lower discount factor. This is because the environment doesn't provide a reward until the episode ends. This reward however is only given to the last action made, therefore the DQN agent has to make actions thinking far into the future as the reward is 0 for all actions except the last one.

A higher learning rate seems to also be beneficial for the learning performance since both the top two graphs have a higher learning rate. It makes sense because a higher learning rate means faster learning but if the rule based agent was a more complex opponent, a lower learning rate would be more preferable since that would make the agent learn more accurately.

It also seems that a lower amount of neurons impacted the performance positively. It's possible to see that the only difference between the dashed graphs is the amount of neurons and the one with the lower amount performs better. The same goes for the dark blue and orange graphs, and even for the green and the red graphs. The only time that this isn't the case is for the top two graphs and it is suspected to be because of luck.

C. DQN vs DQN

For the second method a new DQN agent was created but this time, its opponent was a DQN agent which had surpassed the performance of the rule based agent. The new agent faced its opponent for 90 000 episodes and after every iteration, a new agent was created with different parameters for further testing while the opponent remained the same. This time, only four random combinations of parameters from the first method were tested. The DQN-opponent had been trained against the rule based agent with the same parameters as dark blue in table IV but with 30 000 episodes instead.

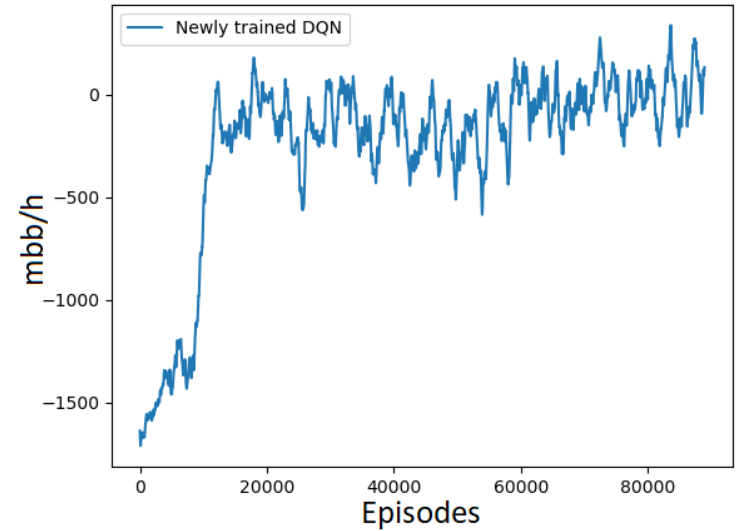


Fig. 5. *mbb/h* for the first DQN agent. Positive value means that the dark blue DQN agent is winning.

Something very apparent with the figures 5-6 and with figure 3 is that DQN seems to have trouble with convergence. For example, in figure 7 and 6, the reward seems to rise in value, just to plummet soon after. This might be because of the nature of poker, that sometimes it is possible to gain a streak of unlucky events. These streaks could be insignificant but it's unsure how many would be necessary. Another theory is that because the action space sometimes changes, (you cannot

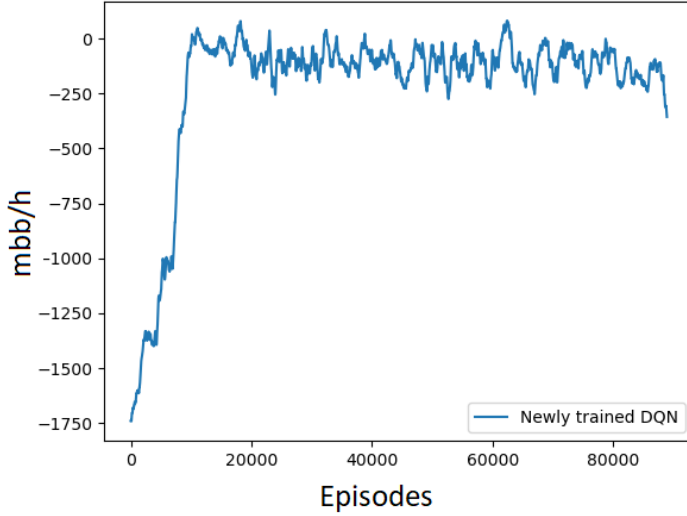


Fig. 6. mbb/h for the second DQN agent. Positive value means that the DQN agent is winning.

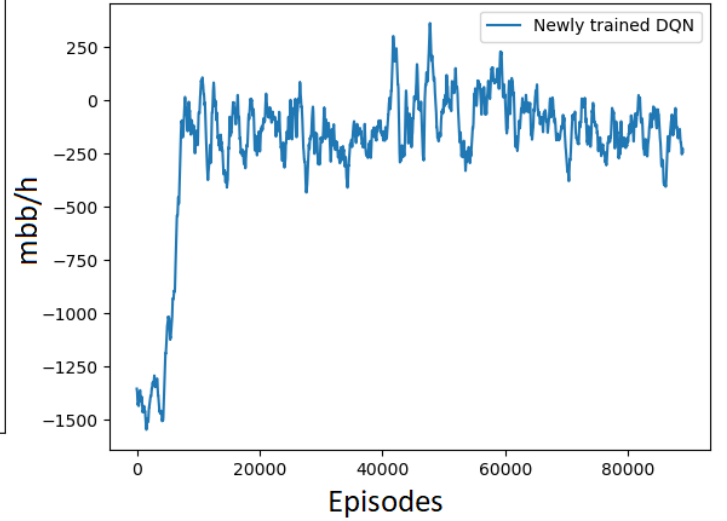


Fig. 8. mbb/h for the fourth DQN agent. Positive value means that the DQN agent is winning.

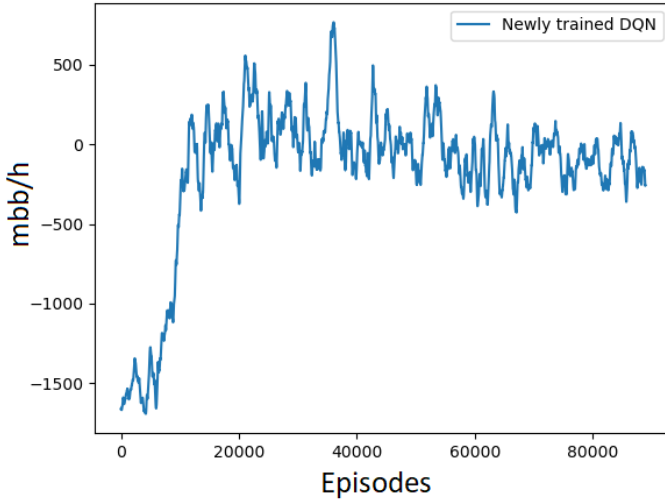


Fig. 7. mbb/h for the third DQN agent. Positive value means that the DQN agent is winning.

always raise or check), the DQN agent needs a lot of episodes to understand why that change happens. It is again unclear how many episodes the agent needs to learn this fact.

D. DQN vs Human

When trying to play against the DQN-agent it was obvious that sometimes it would be stuck at using too simple strategies or stick to only one action constantly. For example, it would often fall into the strategy of always raising. At first it makes sense, if you always raise, you will maximize the amount of money you win. However, this would also maximize the amount of money you lose if you have bad cards. It would also sometimes fall into the strategy of always checking which minimizes the amount of money lost in case of defeat but would also minimize how much money you can win. This is atleast a strategy that beats a random agent but it is not the performance that was wanted nor expected of the agent. It

did however sometimes show some correct behavior such as checking if it had bad cards or raising at the end of the match if it felt it had good cards but these were rare occasions. This correct behavior might show itself more frequently with more training since there are just so many states that the neural network has to approximate the correct action for.

V. DMC & LIMIT TEXAS HOLD'EM

A. Environment

The environment is largely the same as it was in the DQN case. The main difference was that in order to evaluate DMC performance in Limit Texas Hold'em, a DMC agent was trained for 20 millions steps to serve as the opponent. This agent was trained through a simple self-play algorithm, which meant every 50 000 steps the net was saved into a list with a size of ten and for each episode an opponent was chosen at random from this list to serve as an opponent.

The neural network used for DMC was a dynamic net with 3 layers with a size of *Number of neurons*, $2 * \text{Number of neurons}$ and *Number of neurons*. It takes the current observation of the environment and an action and then outputs the Q -value of the action.

To evaluate how a parameter impacted the performance of the DMC algorithm different parameters were varied at time according to table V, the parameters value that wasn't varied can be found in table VI.

TABLE V
VARIED PARAMETERS FOR DMC

Parameter	Value 1	Value 2
Number of neurons	64	512
Discount factor	0.8	1
Learning rate	10^{-2}	10^{-4}

TABLE VI
FIXED PARAMETERS FOR DMC

Parameter	vs Rule agent	vs Pretrained DMC
Number of episodes	50 000	100 000
ϵ	0.50	0.50
ϵ_{min}	0.05	0.05
$\epsilon_{decaysteps}$	30 000 steps	30 000 steps

B. DMC vs Rule agent

The result of DMC when playing against the rule agent can be seen in figure 9 and 10, these graphs are the average of 10 independent repetitions.

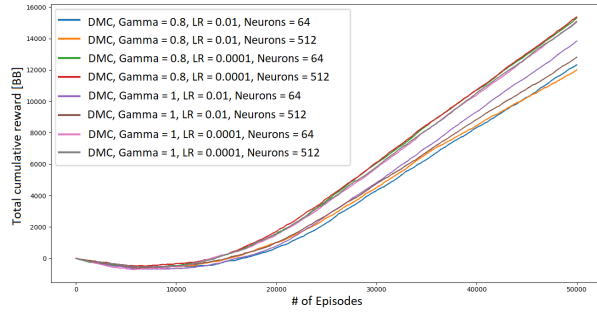


Fig. 9. Total cumulative reward averaged over 10 iterations, with DMC playing against the rule agent.

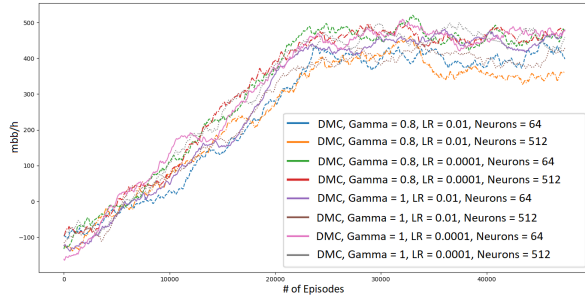


Fig. 10. The mbb/h averaged over 10 iterations, with DMC playing against the rule agent.

According to these results the parameter with the biggest impact on performance is learning rate. This is quite expected and the learning rate is often said to be the one parameter you should tweak if you don't have time to tweak anything else.

However, interestingly a lower discount factor seems to yield a better performance when combined with a lower learning rate. This is quite unexpected since a reward is only given at the end of an episode, one would think that it would be prudent to use a high discount factor to make sure the reward is properly propagated. The reason for this might be because since the reward can get quite high, a lower discount factor might help the algorithm to not get stuck in loops of always raising or folding which can be quite tricky to get out of.

The number of neurons did not significantly impact the performance of the algorithm. This might be because of a combination of lack of episodes meaning that the neurons did not have enough data to properly adjust.

C. DMC vs DMC

The result of DMC when playing against a pre-trained DMC net can be seen in figure 11 and 12, these graphs are the average of 5 independent repetitions.

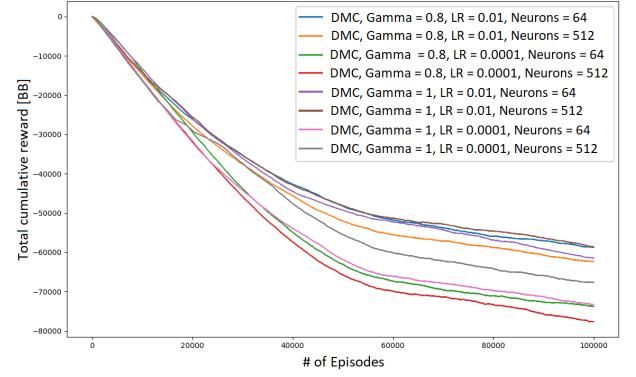


Fig. 11. Total cumulative reward averaged over 5 iterations, with DMC playing against a pre-trained DMC net.

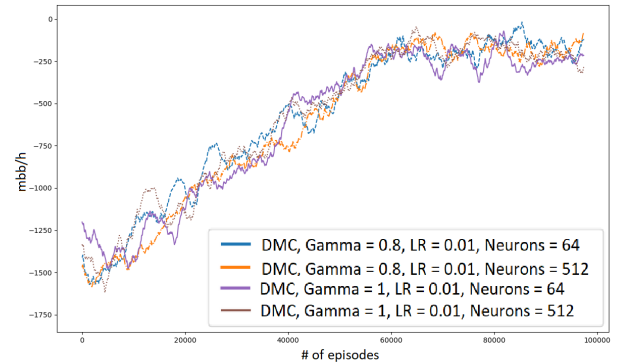


Fig. 12. The mbb/h averaged over 5 iterations, with DMC playing against a pre-trained DMC net for the learning rate fixed to 0.01.

The stark difference when faced with a pre-trained DMC net is that a higher learning rate seems to outperform a lower one, this is contrary to earlier results. However, this can probably be explained by the fact that since the opponent is stronger, a higher learning rate allows the net to learn faster but if we allow the training process to run for longer a lower learning rate might still outperform a higher one.

What is more interesting is that in figure 11 it looks like a lower discount factor is starting to outperform a higher one towards the end of the training. This is inline with the earlier results and as such one could assume that it might be prudent to use a lower discount factor when training neural nets to play Texas Hold'em.

As with before the number of neurons did not seem to impact performance significantly.

D. DMC vs Human

When a human played against the DMC net that had trained for 20 million steps it became apparent that it is unable to outperform humans. While it has learned the value for holding a pair or high cards in its own hand it has yet to learn to use the community cards available to evaluate the hand in most cases. It is also still very rare for the agent to fold even with complete junk cards although it does fold from time to time.

However, despite this it seems to have developed some sort of strategy to raise or call early and later in the round switching to check when it cannot properly evaluate the hand. Meaning that it has at least learned that always raising/checking is not an optimal strategy.

VI. FUTURE WORK

A. DQN

There are some changes that could be done to perhaps improve the agent's performance. One proposal for a change could be to introduce a punishment for using an action too many times in a row. This could force the agent to find a new strategy that doesn't involve just using one action all the time. Another change could be to tweak how often the target network copies the current network as explained in the theory section. It could be that the SGD didn't quite converge when the target network was updated. The last and probably the simplest change is just to try training the agent for a period with a greater amount of episodes. The largest number of episodes used for the agent was 300 000 at the time of writing this report.

B. DMC

For the DMC implementation a few small improvements might yield better results. The most pressing part is the fact that the self-play algorithm simply assumed that after every 50 000 steps the neural network was better, more often than not this was not the case. That meant that even though a neural network was trained for 20 million steps its performance when faced with a human player was rather poor. This could be improved by implementing a scoring system that determines how good a neural network is and only saving the neural network if it outperforms the previous generation. It could for example be based on both the mbb/h and total cumulative reward when faced with each other or be some sort of weighted win rate where depending on the amount bet, it gets adjusted more or less.

There is also the problem that DMC has a high variance, which means that it needs a lot of episodes in order to learn. This could be addressed by implementing parallel processing to allow several different DMC agents to play at the same time and generate data; however, this requires significantly more processing power.

VII. CONCLUSION

In this report we have implemented and compared to popular RL algorithms namely DQN and DMC. With regard to DQN we have shown it can be implemented to solve the Cartpole environment to a high standard.

The DQN algorithm seemed to be mostly affected by the learning rate and discount factor where a higher value for both these parameters was preferable. The algorithm couldn't quite achieve a performance that could be qualified as human but it did show some correct behavior and these behaviors can probably be seen more often with more training.

For the DMC algorithm we found that the learning rate was the most impactful parameter. We also found that a lower discount factor actually helps the algorithm perform better contrary to our prior beliefs. While failing to achieve human performance, the algorithm has shown to have the ability to learn the game and utilize all the different actions at an adequate level.

A future venue of research is to focus on creating a better self-play algorithm for DMC and to introduce a punishment for DQN when raising.

ACKNOWLEDGMENT

We would like to thank our supervisor Alessio Russo for his support and guidance throughout the project.

REFERENCES

- [1] I. Goodfellow, *Deep learning*, ser. Adaptive computation and machine learning. Cambridge: MIT Press, 2016. [Online]. Available: <http://www.deeplearningbook.org>
- [2] R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*, ser. Adaptive computation and machine learning. Cambridge: The MIT Press, 2018.
- [3] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, R. Józefowicz, S. Gray, C. Olsson, J. Pachocki, M. Petrov, H. P. de Oliveira Pinto, J. Raiman, T. Salimans, J. Schlatter, J. Schneider, S. Sidor, I. Sutskever, J. Tang, F. Wolski, and S. Zhang. (2019, Dec.) Dota 2 with large scale deep reinforcement learning. [Online]. Available: <https://arxiv.org/abs/1912.06680>
- [4] 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. (2017, Dec.) Mastering chess and shogi by self-play with a general reinforcement learning algorithm. [Online]. Available: <https://arxiv.org/abs/1712.01815>
- [5] 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 (London)*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.
- [6] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, King's College, Oxford, United Kingdom, 1989.
- [7] J. Chen. (2021, Dec) Neural network definition. Investopedia. [Online]. Available: <https://www.investopedia.com/terms/n/neuralnetwork.asp>
- [8] (2022, Apr) ML Glossary. [Online]. Available: https://ml-cheatsheet.readthedocs.io/en/latest/nn_concepts.html
- [9] D. Zha, K.-H. Lai, S. Huang, Y. Cao, K. Reddy, J. Vargas, A. Nguyen, R. Wei, J. Guo, and X. Hu, "Rlcard: A platform for reinforcement learning in card games," in *IJCAI*, Feb. 2020.

Scalable Deep Reinforcement Learning for a Multi-Agent Warehouse System

Marcus Loberg and Akib Khan

Abstract—This report presents an application of reinforcement learning to the problem of controlling multiple robots performing the task of moving boxes in a warehouse environment. The robots make autonomous decisions individually and avoid colliding with each other and the walls of the warehouse. The problem is defined as a dynamical multi-agent system and a solution is reached by applying the DQN algorithm. The solution is designed for achieving scalability, meaning that the trained robots are flexible enough to be deployed in simulated environments of different sizes and alongside a different number of robots. This was successfully achieved by feature engineering.

Sammanfattning—Denna rapport presenterar en implementation av Reinforcement Learning som löser problemet med att styra flertalet robotar som utför uppgiften att flytta lådor i en lager miljö. Robotarna tar autonoma beslut individuellt och försöker att undvika att krocka med varandra och väggarna av lagerlokalen. Problemet definieras som ett dynamiskt multi-agent system och en lösning nås genom att tillämpa DQN algoritmen. Lösningen är utformad för att uppnå skalbarhet, vilket innebär att robotarna ska vara flexibla nog att agera i miljöer av olika storlek och jämte olika antal robotar. Detta uppnåddes framgångsrikt genom att implementera funktionsextraktion.

Index Terms—Reinforcement learning, Deep Q Networks, Neural Network, Scalability, Multi-Agents, Warehouse Environment

Supervisor: *Hamed Taghavian*

TRITA number: *TRITA-EECS-EX-2022:131*

I. INTRODUCTION

With the field of AI and robotics becoming more sophisticated, human society is now on the verge of a full transformation. Today it can already be seen that autonomous decision making systems are being efficiently implemented in a lot of different industries. The development is rapidly moving towards autonomous cars, automated manufacturing, and even humanoid robots. The spread of autonomous systems will be everywhere and therefore they will have to be designed for coexisting and correctly interacting with each other.

Reinforcement learning (RL) can be applied as a method of designing automated systems. It is different from other branches of Machine Learning because the learning process is not dependent on training data. In RL an agent is defined as an autonomous decision-making system that learns to perform a task by trial and error. The agent observes its environment by analyzing what state it is in, and with that information, it decides what action it should take to reach a new state. With some inspiration from the psychology of animal behavior, a reward system is defined for the learning process. The agent will be rewarded and punished for taking actions that lead

to certain states. All the experience the agent collects, as in states, actions and rewards, is used to form a policy, which serves as the brain of the agent. The agent will change its behavior by updating its policy, with the aim of maximizing collected rewards.

The fundamentals of RL can be traced back to the 1950s when trying to solve the problem of optimal control. It was around this time the renowned mathematician Richard Bellman defined the Bellman equation, describing dynamical systems by states and values, and derived the Markov decision process which makes a subfamily of discrete-time stochastic control processes. These are the concepts that have laid the foundation for all RL algorithms. Today with the increased availability of computational power, the field of RL has become more attractive and a lot of new algorithms and theory is being developed at a rapid pace.

This report presents a scalable solution to the problem of controlling multiple robots that move boxes in a warehouse. The solution is an application of deep Q networks (DQN). The DQN algorithm is a process of learning the environment by mapping specific states to actions by using Q-values. The Q-value is the sum of the immediate reward of taking an action at a state and the discounted future reward, i.e. the return. The mapping is done by using a neural network (NN), which is a function approximator. For DQN, the input to the NN is a vector that represents the state of the environment, and the output is an action-vector with values corresponding to each possible action. The policy of an agent that uses DQN can be described as choosing the action with the highest Q-value for a given state. In the case of the multi-agent problem in the warehouse, the DQN algorithm will be implemented so that each robot can make autonomous decisions individually while working together as a collective.

The method of training the robots was adjusted for achieving scalability. The individual robot was trained so that it could be deployed alongside a different numbers of robots and in warehouses of different sizes. Scalability was achieved by the method of structuring the input state-vector for the NN and implementation of features.

The structure of this report is as follows. A description of the problem and the criteria for the solution are given in Section II. Section III is dedicated to conveying the basic theory of RL, and describe Neural Networks and the DQN algorithm. The method of solving the problem is given in Section IV and the tools used in this project will be given in Section V. Results are presented in Section VI with parameter values used, followed by a conclusion in Section VII and discussion of the results are presented in Section VIII.

II. PROBLEM STATEMENT

In a warehouse, multiple robots will have to learn to perform the task of transporting boxes from one point to another. The challenge is for the robots to perform their task while avoiding colliding with each other and the walls. This problem has to be modeled as a multi-agent system and solved with an implementation of a deep RL algorithm. The project has three main tasks:

- 1) Model the warehouse and encode the tasks robots have to accomplish.
- 2) Develop and implement in each robot a deep RL algorithm.
- 3) Simulate the resulting complex dynamical system.

III. BASIC THEORY

A. Reinforcement learning

The main objective of reinforcement learning is to make an agent learn to perform a task autonomously. Everything outside of the learning agent that the agent interacts with is called the environment and can be modeled as a dynamic system. The agent moves through a sequence of discrete time-steps t , $t = 0, 1, 2, \dots$, during its learning process. At each time-step, the agent receives information of the current state s_t via feedback. The agent uses that information when choosing what action a_t to make. After making the action, the agent reaches the next state s_{t+1} . As an evaluation of the made action, the agent will collect a reward R_t according to a predefined reward system. The agent will be rewarded or punished for taking certain actions and reaching certain states. This process can be seen as an iterative training loop and is depicted in Figure (1).

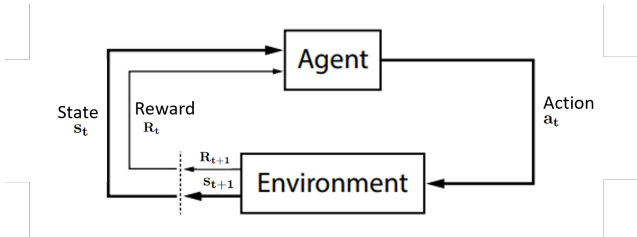


Fig. 1. State-Action loop for an RL algorithm.

The selection of what action to perform was made according to the agents policy π . The policy can be understood as a mapping from each state s and action a to the probability $\pi(a|s)$ of taking action a when in state s . By using the policy, a sequence of actions can be derived that will maximize the received reward. The training process is usually structured into episodes, which start from an initial state t_0 and continues until it reaches a terminal state T . For example the terminal state can be that the agent has finished its task, or reached a limit in the number of steps taken. The learning agent's objective is to maximize accumulated rewards during each episode. It will take an action based on the value of immediate reward R_t and the value of maximum expected discounted reward G_t

for episodic tasks defined as

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \dots \gamma^{T-1} R_T$$

$$= \sum_{k=1}^T \gamma^{k-1} R_{t+k} \quad (1)$$

where $\gamma \in [0, 1]$ is called the discount rate, $0 \leq \gamma \leq 1$. The discount rate decides the value of future rewards at a given state, and is adjusted depending on if the agent should act for immediate reward or act for potential future reward. The experience the agent gains, as in which states it has reached, what actions it has taken and what rewards it has gathered, will be used to update the policy, usually after each time step. The structure of the policy and the method of updating it is defined by the specific RL algorithm implemented to solve a problem. For a more detailed description of the topic of RL see [1].

B. The Markov Decision Process

The previously described concept of the state s_t can be extended to mean any properties of the environment that is observed by the agent. In a real-life example, the states could for a robot be any sensor inputs, which could range from the robot's position to perhaps the humidity of the environment. The agent is able to read multiple state variables at each time-step and make a decision based on all of those. The state variables should for some RL methods be chosen carefully and be relevant to the task the agent is performing. When implementing an RL algorithm to solve a problem the real-life environment needs to be idealized and modeled as a Markov Decision Process (MDP). A finite MDP is defined by a finite state and action space. When defining the environment with a finite MDP property, the state s_{t+1} of the system depends only on the previous state s_t and the action a_t . For an MDP the probability of state s_t and a_t resulting in the state s_{t+1} and the reward r is defined as

$$p(s_{t+1}, r | s, a) = \Pr\{S_{t+1} = s_{t+1}, R_{t+1} = r | S_t = s, A_t = a\} \quad (2)$$

The value of being at a certain state or performing a certain action at a certain state, is estimated through value functions. For MDPs a state-value function $v_\pi(s)$ is defined as

$$v_\pi(s) = E_\pi [G_t | S_t = s]$$

$$= E_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s \right] \quad (3)$$

and gives an expected value of a given state at time step t , while the agent follows policy π . Similarly an action-value function $q_\pi(s, a)$ is defined as

$$q_\pi(s, a) = E_\pi [G_t | S_t = s, A_t = a]$$

$$= E_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a \right] \quad (4)$$

The Bellman equation is a value function that states a relationship between a current state and its subsequent state. It divides the value function into two parts, one for the immediate reward and the value of future discounted rewards. For q_π the Bellman equation is

$$q_\pi(s_t, a_t) = \sum_{s_{t+1}} \sum_r p(s_{t+1}, r | s_t, a_t) [r + \gamma \sum_{a_{t+1}} \pi(a_{t+1} | s_{t+1}) q_\pi(s_{t+1}, a_{t+1})] \quad (5)$$

The Bellman equation is an essential part of most RL algorithms when evaluating values of states and actions.

C. Neural Networks

A deep RL algorithm is signified for using at least one function approximator called neural network (NN). Being inspired by neurological mechanics, the concept of NNs were first created in the 1940s. The use of NNs demands a lot of computational power and with technological development, NNs have become a basic component of modern machine learning. Generally, a NN is constructed by having multiple layers of nodes. See fig (2).

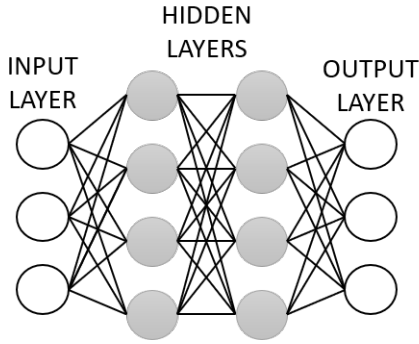


Fig. 2. Schematic sketch of a NN.

The first layer is the input layer, where each input value is inserted in a corresponding node. The layer at the end of the NN is called the output layer and gives the approximated values by this function approximator. The layers in between the input and output layers are called the hidden layers. Except for the input layer, the values of the nodes of one layer are dependent on the values from a previous layer. Each layer can be characterized by the way the values are added together.

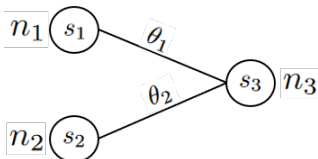


Fig. 3. Connections between two nodes in one layer and one node in the next layer.

Figure (3) above depicts connections between nodes n_i ($i = 1, 2, 3$) from two different layers, and each node has a value s_i . Each connection has a weight θ_i , which is a scalar value denoting the strength of the connection. A type of layer that is commonly used is a linear layer, which sums up the values from previous nodes as a linear combination. In figure (3) the node n_3 is part of a linear layer that sums up the values from n_1 and n_2 according to

$$s_3 = s_1\theta_1 + s_2\theta_2 \quad (6)$$

It is the weights that decide the output of a given NN. The learning process in a deep RL algorithm revolves around tuning the NN by updating its weights for generating a more accurate output.

An additional feature can be added to each layer to make the numerical values of the nodes more manageable. An activation function has the purpose of scaling the value, while still keeping the relationship between the values of a layer. A sigmoid function (7) for example is an activation function, that scales a number to a value between 0 and 1, and is depicted in Figure (4).

$$S(x) = \frac{1}{1 + e^{-x}} \quad (7)$$

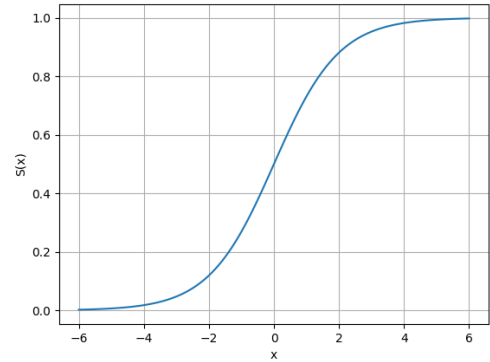


Fig. 4. Plot of the sigmoid function.

Designing a NN can be summarized as choosing the number of nodes in each layer, the number of hidden layers, and what type of layer each layer will be. The process of finding a NN for an algorithm solving a given problem revolves around a lot of testing and is usually not done by an analytical method. To use a NN one could start by looking at similar problems that use NNs, and use a similar NN design as a start point. After that adjust the NN through testing until reaching a satisfying result if possible.

D. DQN algorithm

Ordinary Q-learning is an RL algorithm that learns the values of an action in each state and is independent of a model of the environment. Therefore it is classified as model-free. The method is based on mapping combinations of state variables with possible actions through Q-values (see equation

5). This is done by using a Q-table which is a collection of all Q-values. During the learning process, the values of the table are updated according to the algorithm. The policy π of the agent then states that the action is selected by choosing the action with the highest Q value at a given state.

Deep Q networks (DQN) were first introduced in the paper “Playing Atari with Deep Reinforcement Learning ” in 2013 [2], and is a combination of a Q-learning strategy while taking advantage of NNs. Instead of storing all action-values, the algorithm maps states to action by generating a parameterized value function $Q(s, a; \theta_t)$. The parameters θ_t is the weights for a NN and is updated during the learning process with the use of gradient descent. The observed states of an agent are constructed as an input-vector and are fed into the NN. The output from the NN is an action-vector with Q-values corresponding to possible action. The action with the highest Q-value then becomes the chosen action at a given state. The DQN method relies on equation (8) to update the weights of the NN.

$$\theta_{t+1} = \theta_t + \alpha(Y_t^Q - Q(S_t, A_t; \theta_t))\Delta_{\theta_t} Q(S_t, A_t; \theta_t) \quad (8)$$

where α is the learning rate and decides how large of a step the weights of the NN are updated by. The target Y_t^Q is defined as

$$Y_t^Q \equiv R_{t+1} + \gamma \max_a Q(S_{t+1}, a; \theta_t) \quad (9)$$

The DQN algorithm is signified by the use of a replay memory, where the previous experiences of the agent are stored for some time. The learning process uses the replay memory by randomly sampling a batch of the experiences to use for updating the NN. This is done to remove correlation in the sequence of experiences and smooths the training distribution.

There is a typical dilemma of exploration and exploitation that RL algorithms must take into consideration. During the process of finding an optimal solution, there must be a balance between when to explore the environment and when to exploit the established results, both can't be done at the same time. In Q-learning methods, this is solved by choosing a random action for exploration with the probability ϵ . The learning process starts with a high value of ϵ but will decrease during the process. This will ensure that the agent initially will focus more on exploration but increases focus on exploiting during the process.

This project uses a standard version of DQN which was presented in the paper [3]. It uses two NNs, a primary network Q and a target network \hat{Q} . The target network replicates the primary network every C steps of the episode and is used to approximate the target, i.e expected return, for the next C steps. This will create a delay between the time when Q and \hat{Q} is updated. As a result, this helps with the stability of the algorithm and makes it more inclined to converge.

Algorithm 1 DQN algorithm

```

1: Initialize:
   Total number of episodes E
   Replay memory  $\mathcal{D}$  with capacity N
   Sample size T
   Primary network  $Q$  with random weights  $\theta$ ,
   Target network  $\hat{Q}$  with random weights  $\hat{\theta}$ 
    $\epsilon_{max}, \epsilon_{min}, \epsilon_{decay}$ 
    $\epsilon_{current} \leftarrow \epsilon_{max}$ 
   Max steps M
2: for each episode do
3:   Reset environment,  $done \leftarrow \text{False}$ , Step  $t \leftarrow 0$ 
4:   while  $done == \text{False}$  do
5:      $t \leftarrow t + 1$ 
6:     Observe current state  $s_t$ 
7:     With probability  $\epsilon$  select random action  $a_t$ 
8:     otherwise select  $a_t = \text{argmax } Q(s_t, \theta)$ 
9:     Execute action  $a_t$ .
10:    Observe next state  $s_{t+1}$ ,  $r_t$ , update  $done$ 
11:    Store  $(s_t, a_t, r_t, s_{t+1})$  in  $\mathcal{D}$ 
12:    if stored experience in  $\mathcal{D} > T$ 
13:      sample batch  $(s_j, a_j, r_j, s_{j+1})$  from  $\mathcal{D}$ 
14:      Get target  $y_j$  from  $r_j + \gamma \cdot \max \hat{Q}(s_{j+1}, \hat{\theta})$ 
15:      Perform gradient descent step on  $(y_j - Q(s_t, \theta))^2$ 
16:      with respect to the parameters  $\theta$ 
17:      Every C step set  $\hat{Q} \leftarrow Q$ 
18:      if  $done == \text{True}$  then
19:        break
20:      if  $\epsilon_{current} > \epsilon_{min}$  then
21:         $\epsilon_{current} \leftarrow \epsilon_{current} \cdot \epsilon_{decay}$ 
22:    end while
23: end for

```

IV. METHOD

The algorithm implemented to solve the main tasks of the project was DQN. The DQN implementation used is presented in [4]. In addition to solving the main tasks defined in section II, the multi-agent system presented in this report was also aimed at achieving scalability. This should result in that the agents can be trained in a less complex environment compared to what they will be deployed in. In this project, the complexity of the environment means the amounts of agents, amount of boxes to deliver and the sizes of the warehouses to perform in. Achieving scalability would also result in significantly decreased training time.

A. Environment

The warehouse was simulated as a grid-based environment with multiple robots, boxes, and their pick-up and delivery points. Any box pick-up point can be accessed by all robots and when a box is picked up from a pick-up point it needs to be delivered to a specific delivery point that is a random point on the grid. The robots maneuver through the environment with the possibility of colliding with each other. The number of robots, their start positions, pick-up points, their positions, and the size of the warehouse-grid are set up in three different

complexity levels shown in Table (IV-A). The Delivery goal is the amount of boxes to deliver across all robots.

TABLE I
ENVIRONMENT COMPLEXITY PARAMETERS

Complexity	Robots	Delivery goals	Grid size
1	2	4	7×7
2	8	16	13×13
3	64	128	50×50

In the initial state, the robots start at fixed positions without carrying any box. The episode is complete and the terminal state is met when a set number of boxes have been collectively delivered or maximum steps reached. A limit of 150 Maximum steps is used to avoid infinite loops. The possible actions for the robots to take are moving Up, Down, Left, Right, or Action(picking up or dropping off the box).

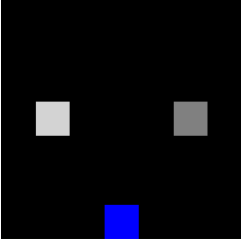


Fig. 5. Complexity 1

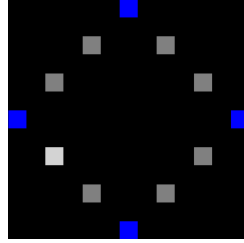


Fig. 6. Complexity 2

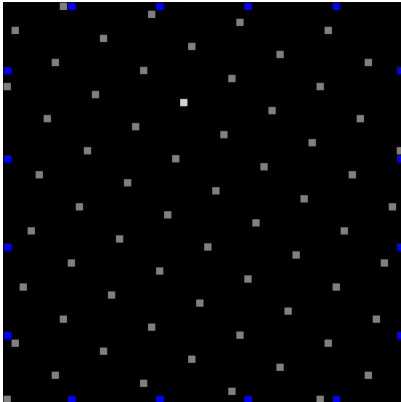


Fig. 7. Complexity 3

The initial states of the graphical simulations are shown in Figures (5, 6, and 7) for the three different complexity levels. The white and grey squares represent the robots (The white being the only robot that updates the NN). The blue squares represent box-pickup points and when a robot has picked up a box there will appear a red box representing the destination.

B. State variables and features

As stated previously, the DQN algorithm uses an input-vector that represents the observed state of the robot. One straightforward approach to constructing the observation vector is combining vectorized matrices that represent the grid environment. The matrices in that project could be one for

representing box pick-up positions, one for the delivery point, and one additional for each robot position. These matrices would be the same size as the grid and would only contain binary information.

This solution is viable but is not optimal to achieve scalability. The size of the input-vector and therefore NN and training time drastically increases for the increased size of the grid environment or number of robots, for complexity 1 with 7×7 grid with 2 robots the size of the input-vector is already $7 \times 7 \times (2+2) = 196$. For complexity 3 this number increases to 165000. Also, the trained model can only work for the exact number of robots and size of grid used for training. These two attributes make the approach inefficient for big warehouses or number of robots because of time and hardware constraints.

A strategy for achieving scalability is to implement the use of features. In Machine Learning features are used for structuring input data for improving the learning process. For this project, features were used to only keep necessary information that is general across different complexity levels of the environment. The feature vector is used as an input-vector for the NN, controlling its size. Through the use of features, the input-vector was made to be 13 binary values for any grid size or number of robots. This should be compared to the standard approach stated before which for complexity 1 had an input-vector of length 196 going all the way up to 165000 for complexity 3. It can be seen that the number of input values has been greatly reduced and the length will remain constant for all complexities. The feature vector for the white robot (which is heading to pick up a box in the destination up and left) in figure (6) is:

$$\begin{aligned} V_{feature} &= [f_1, f_2, f_3, \dots, f_{13}] \\ &= [1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] \end{aligned} \quad (10)$$

The first four values (f_1 to f_4) represent the direction to the closest box-pickup position, in the order of left, right, up, and down. To find what box-pickup position is closest, the distance to each box from every place on the grid is calculated and the number of the box that is closest is saved in the distance-matrix. This is calculated once at the start of the training and used by the robots every time the feature vector is generated. The distance-matrix is as follows for complexity 2 in figure (6).

$$M_{closest} = \begin{bmatrix} 0, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2 \\ 0, 0, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2 \\ 0, 0, 0, 3, 3, 3, 3, 3, 3, 3, 2, 2, 2 \\ 0, 0, 0, 0, 3, 3, 3, 3, 3, 2, 2, 2, 2 \\ 0, 0, 0, 0, 0, 3, 3, 3, 2, 2, 2, 2, 2 \\ 0, 0, 0, 0, 0, 0, 3, 2, 2, 2, 2, 2, 2 \\ 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2 \\ 0, 0, 0, 0, 0, 0, 1, 1, 2, 2, 2, 2, 2 \\ 0, 0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2 \\ 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 2, 2, 2 \\ 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2 \\ 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 \\ 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 \end{bmatrix}$$

f_5 to f_8 represents the direction to the delivery-point, which is a randomized spot on the grid that is updated every time a box is picked up. To give the robots more choice, when the direction is diagonal from the robot both directions are represented as 1s. This applies to f_1 to f_8 .

f_9 is a single binary value representing if the robot is holding a box. The last 4 feature variables, f_{10} to f_{13} are different sensors for evading crashing. A robot left, right, above and below is represented as a 1 in the vector.

Since the vector of features was designed to not be dependent of the size of the grid nor the number of robots, NN size and therefore training time is lowered. Training on very large grids or a large number of robots still takes a lot of time as the simulated episodes to train on will have to be longer. This can be avoided from another property of the features, being that the NN can be trained on a small grid with a few robots and then applied on a larger grid with more robots. This will be tested Test B in the Results.

C. Multi-Agent Implementation in Training

To train the multi-agent system with scalability in mind, each robot uses the same NN. This results in that the amount of NNs does not increase with a higher number of robots. Only one robot samples the state space and updates the NN while the others use the NN to find what actions to take without updating it. This robot that updates the NN is in this report denoted as the main robot/agent and is colored white in the graphical simulation. The training loop is represented in Figure (8).

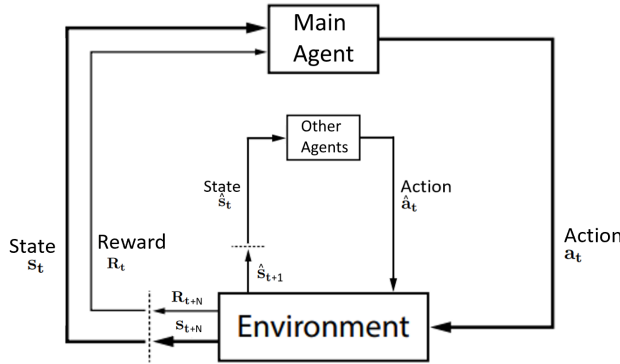


Fig. 8. State-Action-loop for multi-agent implementation

To begin the training, a loop for the main robot/agent is engaged. The main agent observes the environment, receives the features based on the observation and perform an action either randomly or according to the NN following the epsilon greedy policy. After performing the action and updating the environment, it receives a reward and updates the NN. Following this, a loop where the other agents are engaged one by one starts, the loop is identical to the main agent training loop in all but two ways: The actions are only chosen according to the NN and it will not update it, making its reward irrelevant. Although the agent updates the environment. After each other

robot has been engaged the first loop is restarted and the main robot is engaged again.

As every robot uses the same neural network, it is favorable that the main robot experiences each state that all the other robots can experience too. This is done by randomizing what robot is considered as the main one at the start of each episode.

How the rewards are distributed is shown in Table (II), after an action has been made it produces one of six events. These events give rewards valued at how much the algorithm should strive to do or avoid them. Two different reward tables are used, the second one was made to further reduce crashes for the second test specified in the results.

TABLE II
REWARDS FOR EVENTS

Event	Reward A	Reward B
Delivering box	10	20
Picking up box	5	10
Move	-1	-1
Action not possible	-3	-3
Crashing into wall	-5	-5
Crashing into robot	-30	-20

Each move taken gives a -1 reward, this is to incentive taking the shortest path. Action not possible is when trying to pick up when there's no box there or dropping off when the robot doesn't hold a box.

The features make it possible for the robots to learn general knowledge, that can be applied to more complex environments. For instance, the neural network is not dependent on the size of the grid nor the number of robots, this means we can train on a small grid with a small number of robots and then apply it on a huge grid with much more robots.

V. TOOLS

The coding for this project was done using python. The main modules used for the environment during the training phase was *NumPy*, and the algorithm was implemented with a combination of both *NumPy* and *Pytorch*. The graphical simulation module was *PySimpleGUI*. Full list of used python modules:

`NumPy`, `Torch`, `IPython.display`, `Random`, `Collections`, `Matplotlib`, `Copy`, `PySimpleGUI`, `Time`

VI. RESULTS

This section serves to present the results of implementing DQN with features in the warehouse simulation, first with training done in the same complexity of environment as deployed in for all three complexities (Test A). A second test (Test B) is made for training in a set training environment before implementing in the three different complexity environments. An effort to compare the performance between these implementations are made. The following result in this section was based upon DQN parameters in Table (III) and NN dimensions in Table (IV), which was chosen through testing.

TABLE III
TABLE OF PARAMETERS USED FOR TRAINING

Variable	Episodes	ϵ	γ	α	ϵ_{decay}
Complexity 1	2500	0.9	0.95	10^{-4}	0.9999
Complexity 2	2500	0.9	0.95	10^{-4}	0.9999
Complexity 3	4000	0.9	0.975	10^{-4}	0.99997
set training env	4000	0.9	0.95	10^{-4}	0.99997

TABLE IV
PARAMETERS USED FOR THE NEURAL NETWORK

Layer	Nodes
Input	13
Hidden #1	75
Hidden #2	50
Output	5

Both of the hidden layers in Table (IV) are linear layers with a ReLu-activation function. The output layer is a linear layer without an activation function

A. Training in the same environment

By training the model on the respective complexity environment using the reward table Reward A in Table (II) and deploying it on the same environment the results in Table (V) is achieved. Reward are from getting the reward of the main robot after each episode, crashes are the total number of crashes between robots for all robots combined during the episode and success rate is the percentage of episodes were the desired amount of boxes is delivered. These results are averaged across 20 episodes.

TABLE V
RESULTS DEPLOYING DQN MODEL

Complexity	Reward	Crashes	Success rate	Training Time
1	-16.8	0.8	100 %	3.4m
2	-56.5	1.6	100 %	7.6m
3	-1672.8	51.2	0 %	86m

The plots (9 to 11) show the training curve for complexity 1 to 3 respectively, The reward is averaged over 10 episodes.

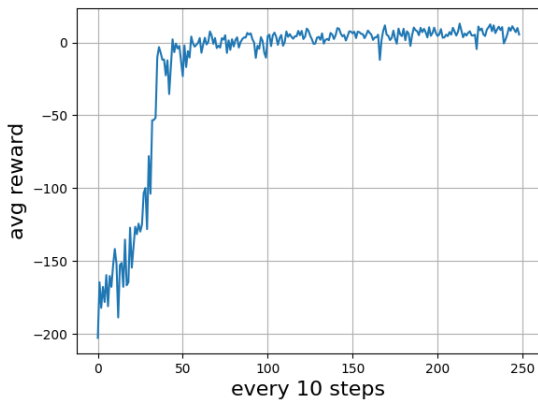


Fig. 9. The training curve for Test A complexity 1

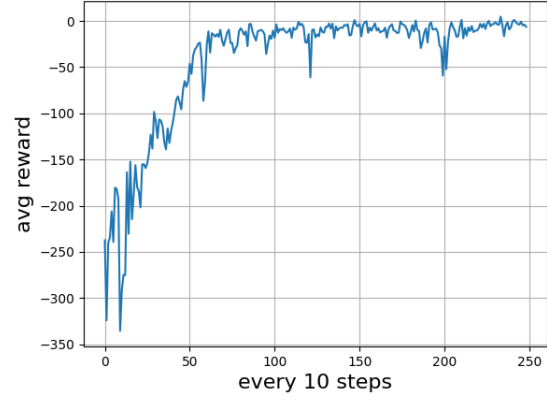


Fig. 10. The training curve for Test A complexity 2

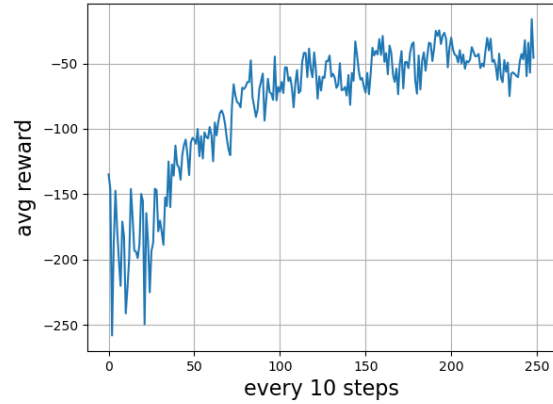


Fig. 11. The training curve for Test A complexity 3

B. Training done in a set training environment

The following result is an attempt of using a dedicated training environment to train the model on and then deploying the model in the three different complexities of environments. This is a challenge as the model was never trained in the states it will observe in the deployment. The model will have to learn general information that applies to different environments never seen before. The dedicated training environment is made as a 5×5 grid with 5 robots, the initial positions of these robots are randomized every episode. The delivery goal is 15 and reward table Reward B in Table (II) is used. An example of how the initial step looks is shown in Figure (12).

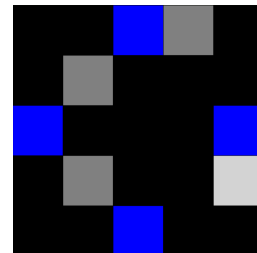


Fig. 12. The initial state in the training environment for Result B

The parameters used during training are shown in reward table Reward B in Table (III). The plot in figure (13) shows the reward converging in the training curve, The reward is averaged over 10 episodes.

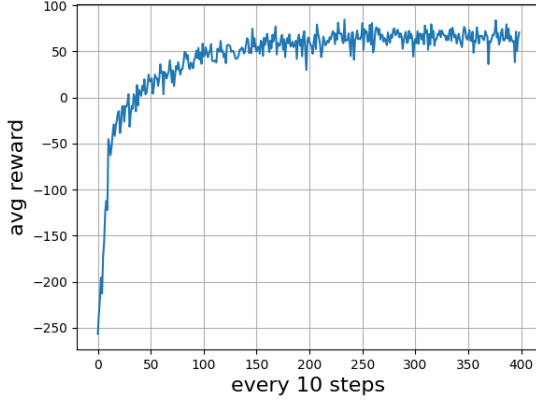


Fig. 13. The training curve for test B

The result after training this model once and deploying it in the 3 complexities of environments are shown in Table (IV).

TABLE VI
RESULTS DEPLOYING DQN MODEL

Complexity	Reward	Crashes	Success rate	Training time
1	43	0.3	100 %	7.4m
2	26	1.6	100 %	7.4m
3	-120	14	100 %	7.4m

The following frame sequences (14) through (19) show 6 sequential images during the middle of deployment of the finished model in complexity 2.

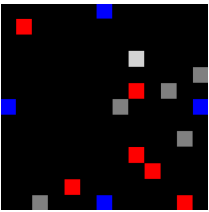


Fig. 14. Frame 1

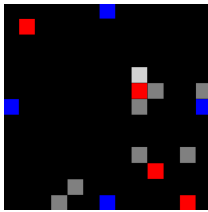


Fig. 15. Frame 2

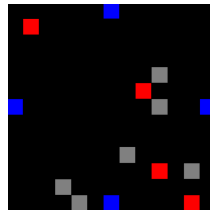


Fig. 16. Frame 3

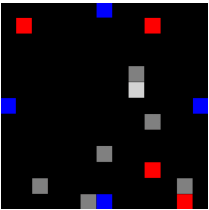


Fig. 17. Frame 4

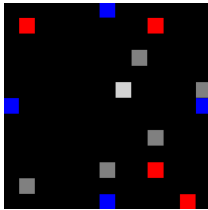


Fig. 18. Frame 5

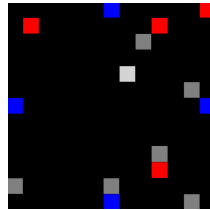


Fig. 19. Frame 6

VII. DISCUSSION

The results from Test A show that solving the problem by training in the same environment as it's going to be deployed in works well for simpler environments. When increasing the size of the grid and number of robots (through increased complexity) the training time increases while performance declines. The performance decline could be because with increased grid size, the agents will have to perform many steps before getting a reward, making the learning process harder. The increased number of robots will increase both training time (as more robots have to perform actions during training) and crashes as more robots can crash. All this combined led to a 0% success rate, many crashes, and a large training time in Complexity 3. Although the parameters used for training were sub-optimally tuned, optimal parameters will increase performance but training time will still be large in comparison to training in smaller environments as presented in test B.

The parameters and reward table for test A should be optimally tuned across all complexities further to achieve better performance and fewer crashes. Although especially in complexity 3 as the Training curve doesn't converge as clearly and the performance was unsatisfying.

In test B the results show that training in a smaller environment and then deploying in the desired environment can increase performance compared to using the same environment as in test A. The results show that the number of crashes decreases, the success rate improves and training time is low. The training time is static as the results were achieved by training the model once and applying it in the three different complexities. This combined with the results being good across all three complexities show that scalability was achieved. To further motivate scalability being achieved, there is nothing in the way of deploying the trained model in a much more complex environment than the one presented in this report.

Note that the reward in the results should not be compared across tests A and B as the reward tables are different across the tests.

The number of crashes is still high across both test A and B but it can be seen in the image sequences (14) through (19) that the robots have learned to try and avoid crashing. In (14) the white robot is heading down to deliver a box at the delivery point below it, at the same time the grey robot to the right of that delivery point is heading left. During the following images (15) through (16) it can be seen that the grey robot moves up to avoid crashing into the white robot.

The reasons for the high number of crashes can be a result of a lot of different reasons, Firstly the parameters are sub-optimally tuned because of time constraints. Some other reasons are presented under.

The features the group has presented may not be enough for the robots to be able to completely avoid crashing. With the features used in this project, there is no way for the robots to plan ahead as they can only observe robots that are beside them. Further experiments to add more features to help the robot to avoid crashing might be needed.

The decision to use one Neural network to control all robots might also be a problem as this might limit the robots from

learning specific policies that help them for the exact position they start at. Using multiple NN might decrease crashes in our tests where the delivery goal is small and initial states play a role in crashes but it won't help with real-life applications where there is no delivery goal.

Another issue might be the training environment used for Test B. In this project, it was designed to be small with a lot of robots to give the robots a lot of situations where crashes are present. It was also designed to have all possible observations that might be encountered in the three complexities. The number of robots, the size of the grid, or the delivery goal can all be changed and might result in better performance.

The Reward tables used can also be changed to give better performance. Increasing the negative reward for crashing is the intuitive solution to decrease crashes. The problem with this is that if set too high, the robots learn to walk back and forth on the spot as this gives a higher reward than trying to deliver boxes which sometimes leads to crashes.

Additionally the human factor in the role of coding errors can be the reason for the high number of crashes.

VIII. CONCLUSION

DQN with features shows great promise in being used for solving dynamical multi-agent optimization problems efficiently. Even with sub-optimally tuned parameters, The results from both tests show that our implementation works. Comparing results from test A and test B show that features can be used to train the model in a different environment as to deploy in, decreasing training time and increasing performance, especially for complex environments. The problem statement in the report was solved and the idea of achieving scalability was successful.

ACKNOWLEDGMENT

The authors would like to thank the project supervisor Hamed Taghavian for his feedback and support.

REFERENCES

- [1] R. S. Sutton and A. G. Barto, *Reinforcement learning : an introduction*, ser. Adaptive computation and machine learning series, Cambridge, Massachusetts, 1998.
- [2] 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>
- [3] 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 (London)*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.
- [4] A. Zai, *Deep reinforcement learning in action*, 1st ed., ser. Manning Publications, Shelter Island, NY, 2020.

Context D & Context E

D: Embedded systems and motor drives for electric transportation

E: Semiconductors for Embedded Systems

POPULAR DESCRIPTION

What's that in your pocket?

Probably a phone. How long has it been since you were more than one meter away from an electronic device? One day? One week? One year? You are probably reading this article on your smartphone right now, but have you ever wondered what enables our phones to make calls, our cars to run and our coffee machines to brew coffee? All these devices have one thing in common: they rely on embedded systems.

Embedded systems are chips consisting of computing hardware and software, with the purpose of solving a specific task. They are used in all kinds of electronic devices, such as the camera on your phone or Face ID. The building blocks of these systems are transistors, tiny switches that can be turned on or off. The latest iPhone 13 has an incredible transistor count of 15 billion! To put that number into perspective, imagine a single blade of grass on a football field corresponds to 1 transistor, then it would take 30 football fields to make an iPhone 13.

In the future, as embedded systems continue to evolve and become present everywhere the following might occur: you are driving to work and your smartwatch recognizes that you are tired and therefore starts brewing coffee at your workplace. Suddenly you suffer a heart attack, but electrical devices on you recognize this and order your car to drive to the nearest hospital as well as notifying your boss about this. Smart connected devices containing embedded systems will make your life more convenient and safer in unforetold ways.

Additionally, embedded systems also have the potential to dramatically decrease carbon emissions since most, if not all, alternative energy sources such as solar and wind are heavily dependent on embedded systems. Without them, connecting these sources to the electric grid would be impossible. No doubt, a society independent of fossil fuels can only be made a reality with the use of embedded systems. Therefore, research in this area is crucial for the transition to a sustainable tomorrow.

SUMMARY OF PROJECT RESULTS

The complexity of vehicles in the transportation industry has increased over years, with the demand for more functionality and energy efficiency. In order to achieve this, embedded systems are used primarily for communication and regulation purposes. These systems can be used to determine the remaining charge in a battery, monitor temperatures in a vehicle, regulate the speed of a motor or analyze data. Today's embedded systems are based on semiconductors, but in the future, they could potentially be replaced by more energy efficient alternatives such as spintronic technology.

All projects in context D were conducted as part of student competitions among teams from different universities. Projects D1, D2, and D3 were conducted within the KTH Formula Student (KTHFS) team. KTHFS is a student only racing team that builds an electric formula (open wheel) race car to compete against other universities. Almost everything is designed and manufactured by the team members themselves. This year's car is called DeV17 which stands for "driverless electric vehicle

17", since the car is electric and can be driven by a student from the team or run in self-driving mode. Project group D5 was a part of the so-called KTH Delsbo Electric team. In Delsbo Electric, student teams from different universities compete in making the most energy efficient light rail vehicle. All vehicles that participate during the competition are battery driven and take six passengers. Delsbo Electric has been held yearly since 2002 but KTH has yet to be represented in the competition. This year, a number of students hoping to change that have put together a team and started developing a vehicle, aiming to participate in the competition held in May.

Project group D1 developed the software platform for the Accumulator Management System (AMS) for the powertrain of the DeV17 electric race car. The AMS system monitors the total current, cell temperatures and cell voltages of the accumulator, ensuring that they do not operate in critical conditions according to the cell manufacturer's specifications and in compliance with the Formula Student Germany rule book. The safety critical tasks are proven to execute within the specified time frames, while allowing the processor to simultaneously handle peripherals such as Analog-to-Digital Converter (ADC), General-Purpose Input/Output (GPIO), Controller Area Network (CAN) and Isolated Serial Peripheral Interface (isoSPI). It also allows for continuous real-time estimation of the State of Charge (SOC) and State of Health (SOH) of the accumulator. This necessitates the use of a Real-Time Operating System (RTOS) based software platform so that different tasks can be executed simultaneously. Accurate estimation of SOC enables increased range/mileage of the vehicle, while an accurate SOH is crucial for guaranteeing a safe operation of the vehicle.

In the future, more cutting-edge and experimental SOC/SOH estimation algorithms, perhaps incorporating embedded neural networks) may be tested and deployed on the working AMS software platform, to further bridge the gap between theoretical work and practical experimentation in this subject.

Project group D2 constructed hardware and software for a Charge Controller (CC) for the DeV17 vehicle. The purpose of the CC was to enable easier and safer charging of the vehicle battery. The hardware consisted of a custom-designed Printed Circuit Board (PCB) which had a Microcontroller Unit (MCU) at the heart of the CC, specifically a STM32F769IIT6 was used. The software written in C consisted of a Graphical User Interface (GUI) and logic for handling communication. In conjunction with the hardware and software, the CC was able to communicate with other systems of the vehicle, primarily with the AMS mentioned in project D1. The communication protocol used was CAN, which has been used widely in the automotive industry because of its noise-tolerant properties and low cost. The CC has provided a safe, user-friendly product for charging the vehicle's batteries.

In the future, more features could be added to the CC, such as connecting thermistors in order to monitor critical components. One such critical component is a diode that all charging current flows through which means it is important that it does not overheat. In regards to software, additional code can be written that provides the user with real-time cell temperature data acquired from the AMS as well as displaying data from the energy meter used within the battery.

Project group D3 constructed a prototype version of a Data Acquisition Unit (DAU) for the DeV17 vehicle. The purpose of this unit was to gather data from sensors in the vehicle. In order to do this, an analog backend was built, which consists of a wheatstone bridge, an Instrumental Amplifier (IA) and an active low pass filter together with a MCU on a PCB. In the analog circuit, digital potentiometers were used to balance the wheatstone bridge and adjust the gain in the circuit to add flexibility, since different types of resistive sensors should be compatible with the unit. The MCU regulated the digital potentiometers using inter-integrated circuit (I²C) serial communication bus. It also converted the analog signal from the analog circuit to a digital signal using internal ADC channels, which could then be displayed on a computer. In order to measure the circuit's performance, different tests were conducted to characterize its accuracy, precision, resolution, and dynamic range.

In the future, more types of analog backends can be investigated to add more generality and flexibility to the data acquisition unit. A general purpose DAU can minimize the number of circuit boards needed in the car, and therefore save components and minimize its carbon footprint.

Project group D5 developed a driving cycle, a drive system and a control system for an electric light rail vehicle participating in the Delsbo electric competition. The goal was to provide a working vehicle and minimize its energy consumption by optimizing the driving strategy for the competition track. To reach this goal, a simulation of the vehicle and track was

developed in Simulink and used to examine different driving cycles. The final driving cycle combines careful uphill acceleration with the “Pulse and Glide” strategy on the level part of the track. Net energy consumption is further reduced by the use of regenerative braking downhill. Drive system parameters for the motor, battery and motor controller were determined by simulating a variety of different driving strategies. Furthermore, the aim was to design a control system that combines manual and automated driving. It should be written on a microcontroller controlling the motor speed via a driver.

This project has created a general basis for future optimization work. It would be of interest for future projects to explore the possibilities of optimizing the electronic system from an operational standpoint since the operational power consumption has not been considered as part of the current scope.

Project group E2 studied an alternative embedded memory technology for microcontrollers called embedded Magnetoresistive Random Access Memory (eMRAM). Today’s microcontrollers use Static Random Access Memory (SRAM) and flash memory, which are both based on semiconductors. eMRAM, on the other hand, is based on spintronic technology and uses a memory storage element technology called magnetic tunnel junction. eMRAM is smaller and more energy efficient than both SRAM and flash memory, with equal or greater operation speed. The project showed that eMRAM could replace existing semiconductors based memory in microcontrollers to improve them.

Future projects could focus on other emerging memory technologies such as Resistive RAM (RRAM) or metal oxide resistive RAM (oxRAM). These are both memory technologies proposed to enable artificial intelligence and to investigate how such memory types could make this possible.

IMPACT ON SOCIETY AND ENVIRONMENT

Our modern society is heavily dependent on embedded systems. They are essential for commercial products such as smartphones, computers, TV et cetera. They are also important in different areas in society for example in health care and transportation.

Embedded systems today are highly dependent on semiconductors. These are made of rare-earth elements, which are often mined under poor working conditions in a way that damages the local environment. Furthermore, the production process requires extreme amounts of water and energy and produces considerable amounts of hazardous waste. In contrast, the final product is very energy efficient. A CMOS transistor for example, theoretically only leaks current during switching. And yet, the amount of current losses in the switching of all CMOS in the world is massive. Therefore, it is of importance to research alternative methods such as the spintronic technology, which does not need an electric current to switch states. Using spintronic technology in embedded systems would decrease energy consumption and benefit the environment.

As society becomes more electrified and aims to decrease emissions of CO₂, the demand for green electricity increases rapidly. If all current transportation was to be electrified today, renewable sources would not be enough to meet the increased energy demand. Fossil fuels would have to be used to produce electricity, which would go against the main reason for using electric transportation in the first place.

Production of environmentally friendly products could have a rebound effect. For example, consumers might wrongfully think that they are doing the environment a favor by switching to an electric car even though their current fossil fuel car still works. Even if electric cars do not contribute to CO₂ emissions during driving, production and transportation to the consumer has a large ecological footprint, especially the batteries. They contain lithium and cobalt, which are also often unethically mined and hard to recycle. Consumers might also utilize their new cars more since they are supposedly environmentally friendly, leading to increased traffic and energy usage. This would ultimately result in a net negative impact on the environment. This is something that consumers need to be aware of.

Although great progress has been made within the field of electric transportation in recent years, there are still factors that make them unattractive to customers. For example, petrol can be stored for emergencies but if an electric car discharges out of reach of a charging station, there is practically nothing one can do about it. Another problem is that short range and lack

of access to charging stations make them unsuitable for many people living in rural areas. Infrastructural changes are needed to make electric vehicles accessible to everyone.

Since integrated circuits are becoming smaller it is possible to house more of them in an embedded system of a given size. This enables the possibility for more features for the user, but it also leads to concerns regarding privacy. The reason for this is that when systems become more complex it also becomes harder for the average individual to understand them. This gives other parties the opportunity to exploit the individual by integrating functions into the system that the user is not aware of.

In conclusion, embedded systems have a large impact on both society and the environment. They have made technological advancements possible in numerous fields, but their complexity brings integrity related problems. While enabling more environmentally friendly solutions, they also have a negative impact on the climate. Embedded system engineers need to take all these aspects into account early on when designing products. The projects from contexts D and E all have the possibility to have a positive impact on the climate, but it depends on how they are used.

Battery Management System Software for a High Voltage Battery Pack

Emil Tagesson and Oscar Eriksson

Abstract—The electric vehicle industry is experiencing a boom in funding and public interest, and the formula student movement is following suit; an electric race car is currently being developed by the KTH Formula Student organisation (KTHFS) which is the cause of this work.

Consumers desire increased speed and range, and are unwilling to compromise one quality for the other. This necessitates the use of lithium ion cells, which may explode and exhume toxic gases if over-strained with respect to current, charge or temperature. A robust, reliable and provably safe battery management system should therefore be developed. There are numerous methods to further increase the mileage to get an edge on competitors, such as cell balancing and live estimation of the State of Charge (SOC). It is also vital that old and/or deteriorated cells should be identified and disposed off in due time, and State of health (SOH) estimation provides a means to do this. In this paper a complete battery management system software solution is developed and presented, utilising methods like simulation and code generation to create a program that runs on a real time operating system (RTOS). Some real world test were conducted and some results are simulated. The finished BMS performed well in tests, meets all goals and meets all timing constraints. The project can therefore be considered as successful.

Sammanfattning—Intresset för elbilsindustrin har på sistone vuxit något markant, och formula student-rörelsen har anpassat sig efter dessa trender; en elektriskt bil tillverkas just nu av KTH Formula Student organisationen (KTHFS) vilket ger upphov till detta arbete.

Marknaden vill ha snabbare bilar som dessutom har förbättrad räckvidd, men vägrar offra den ena egenskapen för det andra. Lösningen är att använda litiumjonceller. Dessa har dock en säkerhetsrelaterad nackdel; om cellerna utsätts för alldeles för höga eller låga temperaturer, strömmar eller laddningsnivåer kan de explodera och utsöndra giftig gas i luften. Därför är det lämpligt att skapa ett batterimoniteringssystem vars funktion och säkerhet kvalitativt kan utvärderas och bevisas. Det finns flera metoder för att få förbättrad prestanda ur sin ackumulator (batteriensemble); cellnivåbalansering och laddningsnivåestimering (SOC) implementeras i detta projekt. Föråldrade/utslitna celler bör identifieras och avskrivas i god tid. Celldeklineringsestimering (SOH) är ett sätt att lösa detta problem. I denna rapport presenteras en fullständig implementation av mjukvaran för ett batterimoniteringssystem, där metoder som kodgenerering och simulering utnyttjas för att skapa ett program som kan köras på ett realtidsoperativsystem (RTOS). Vissa test gjordes i verkligheten och vissa resultat simulerades. Det färdiga batterimoniteringssystemet presterade väl i test, uppfyllde alla mål samt mötte alla tidskrav. Projektet kan därför anses som lyckat.

Index Terms—Battery Managment System, State of Charge, State of Health, Real Time Operating System, Cell Balancing, Code Generation

Supervisor: *Matthias Becker*

TRITA number: *TRITA-EECS-EX-2022:135*

I. INTRODUCTION

At the heart of every battery electric vehicle there is a safety-critical system which continuously monitors battery current draw, state of charge, state of health, capacity, insulation, cell voltages and cell temperatures. The KTH Formula Student race car DeV17 uses lithium ion cells, a cell type which is known for having among the best available power density at the given price point, but which can potentially ignite and consequently exude toxic gas if improperly strained to a point of catastrophic failure [1].

Such a system is often called a battery management system (AMS). In the case of the DeV17 race car most of the AMS functionality is implemented in software on a single processor microcontroller. Because of the multitude of simultaneous objectives, a multi-tasking framework known as a Real Time Operating System (RTOS) is employed. RTOS-based programs are uniquely suitable for solving *hard* time constraints, time constraints such that if they are not met will result in potentially catastrophic consequences [2].

Batteries are expensive and heavy. To optimise the speed and range of the race car, one should maximise the amount of charge that might be withdrawn in a drive cycle while guaranteeing operation within safe predetermined bounds. This is done using several strategies such as cell balancing, state of charge estimation and state of health estimation, which are discussed and implemented.

For the DeV17 car to be eligible for the Formula Student Germany (FSG) 2022 competition, it has to pass numerous criteria, many of which apply to the AMS. These come in the form of time constraints, software support for connected peripherals as well as computer connectivity and data presentation. These can be found in the formula student Germany rule book [3].

LIST OF ACRONYMS

TS Tractive System
AMS Battery Management System
CAN Controller Area Network
RTOS Real-Time Operating System
AIR Accumulator Isolation Relay
isoSPI Isolated Serial Peripheral Interface
SC Shutdown Circuit

SOC State of Charge
SOH State of Health
OCV Open Circuit Voltage
MC Micro Controller
KTHFS KTH Formula Student
ADC Analog-to-Digital Converter
ISR Interrupt Service Routine
ECM Equivalent Circuit Model

II. HARDWARE DESCRIPTION

The full hardware ensemble is referred to as the tractive system (TS). It is housed in an aluminium casing with an electric fan attached to it (which is operated by the AMS program), and contains the following components:

A. Accumulator

The accumulator is a collection of 6 serially connected segments. These segments are collections of 21 serially connected cell groups. The cell groups contain two parallel battery cells of the type specified in [4]. The maximum total voltage of the accumulator is 528 V, and the maximum output is set to 50 kW.

B. BECKY

Becky, pictured in Figure 1, is an ensemble of many printed circuit boards with different purposes. A circuit model illustrating the working principle of most relevant electrical elements is pictured in Figure 2.

1) *AMS-slaves*: There are in total twelve AMS-slaves, two mounted on every segment, each containing an LTC6804 chip [5]. These perform 18 voltage measurements using an analog-to-digital voltage converter (ADC); twelve cell voltage measurements, five auxiliary measurements and one voltage reference measurement. The auxiliary measurement points are connected in thermistor circuits using the aforementioned voltage reference, so that the auxiliary voltages correspond to five temperatures.

2) *IVT*: The IVT measures the accumulator current and voltage. It also measures the vehicle side voltage, meaning the charger or inverter depending on the hardware configuration.

3) *AMS-master*: The AMS-master has a STM32F407VG micro controller (MC) [6] which runs the main program, and an LTC6820 chip [7] for serial communication with the AMS-slaves. There is hardware support for JTAG (joint test action group, used for uploading code/debugging), CAN (controller area network, main means for communication between devices) and isoSPI (isolated serial peripheral interface, used for communication with the AMS-slaves) connectivity.

The STM32F407VG features an ARM-Cortex M4 Core with an operating frequency of up to 168 MHz. It has up to 1

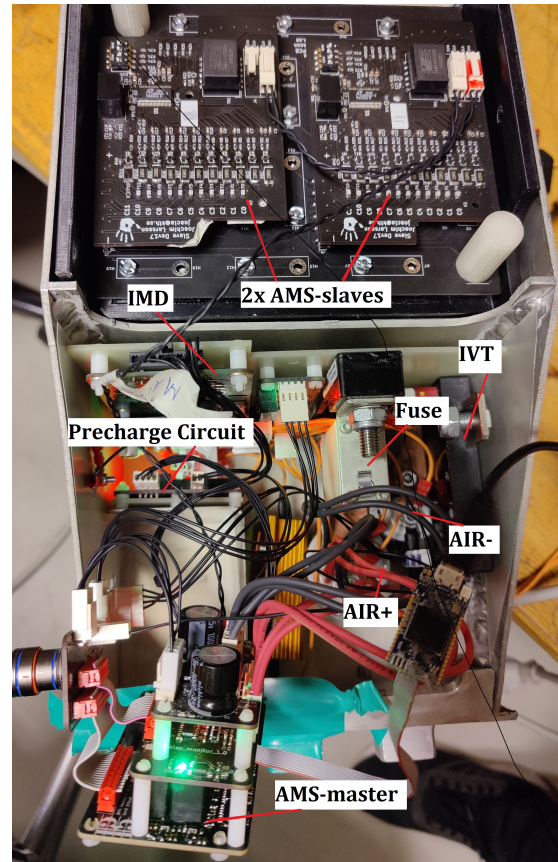


Fig. 1. BECKY; all of the components in depicted in 2, as well as the IMD and chassi.

Mbyte of Flash memory and up to 192 KBytes of static RAM.

The board also hosts two relays controlled by the AMS- and IMD error signals respectively. The relays are featured in the shutdown circuit (SC) that is illustrated in Figure 3. These can be opened by the AMS-master, but can not be closed by the AMS-master. These instead have to be opened by an electrical systems officer (ESO) who sends a resetting signal (by pressing a physical button mounted on the vehicle). An ESO is the only person who can declare the car electrically safe, and work on the electrical systems during competition. [3].

4) *AIRs*: Becky features two accumulator isolation relays (AIRs) that make up the accumulator current path. These are referred to as the positive side AIR and the negative side AIR. No current may flow through the cells if either one of the AIRs are open, except for any individual cell current due to intentional cell balancing functionality. The AIRs are powered by the SC signal, pictured in Figure 3, and as such can be powered down by setting the AMS or IMD error signals. The TS system is then considered as being shut down.

5) *IMD*: The IMD supplies the AMS with two signals; the IMD_ok which is low when any fault condition is triggered such as a device error or an insulation level violation between the vehicle side of either AIR and the chassi. The

The schematic diagram illustrates the LV supply system architecture. At the top, the 'LV Supply' provides power to a series of components: LVMS, BSPP, IMD, AMS, Cockpit, Left, and Right Shutdown Buttons. These components are connected to a main power line. A '100 mm max.' dimension is indicated for the cable length. The main line passes through an 'Overcurrent Protection' unit and an 'LV Battery'. A secondary line branches off to TSMS, BOTS, Inertia Switch, RES, and AS. A central 'Interlock' unit is connected to the main line and the secondary line. The system also includes optional Precharge Circuitry, EBS Relay Coil, AIR Coils, and Activation Logic. A legend indicates that a circle with a diagonal line represents a 'normally closed Element' and a circle with a diagonal line and a dot represents a 'normally open Element'.

IMD also has a signal which encodes the insulation level which may be used to validate the AMS-masters functionality.

III. PROBLEM FORMULATION

The stakeholders are satisfied when all applicable FSG rules are complied to. As follows are all applicable FSG rules [3]:

- FSG Rule 3 (EV 5.8.11), It should be possible to view all cell voltages and temperatures on a graphical user interface.
- FSG Rule 4 (EV 6.3.3), Within 30 s of the insulation level being set to $\leq 250 \text{ Ohm} / \text{V}$ the IMD should trigger an error.
- FSG Rule 5 (EV 7.1.5), The AMS should be able to shut down the charger if a critical error is detected in the TS.

- R1, the AMS must permit three modes of operation:

- The various precharge procedures simply involves connecting the negative side AIR, followed by the precharge relay (that sits in series with a resistor). The vehicle voltage is then allowed to reach 95% of the accumulator voltage (per rule EV 5.7.1). This is followed by connecting the positive side AIR and then disconnecting the precharge relay/resistor. When charging the charger must first be turned on, and the voltage/current thresholds must be set. Precharging is done to limit inrush currents since the voltage difference between the charger and accumulator or accumulator and inverter can be several hundred volts.

IV. THEORY

A. RTOS

Since the AMS is a uniprocessor system, calculations can't run simultaneously. There are a lot of tasks that need to be done within a given time constraint. Having a kernel that can determine what task that needs to be executed and when (often called scheduling) is a great advantage and allows for efficient use of the processing time available. Being able to run tasks in this way is called running concurrently, since the kernel allows for pausing of a task with less priority then executing one of higher priority to later jump back into the one with less again. Such a system is called a real time operating system, or RTOS. Real time doesn't necessarily mean that it's fast, more that it's deterministic with respect to tasks being executed within a given time constraint [9, p. 1-7].

The AMS can be considered a hard real time system, because of the dangers of electrocution in the case of compromised insulation, or cells exploding when over-strained. Each task has a specified deadline within which it has to produce an output or respond to an external input. If any deadline in a hard real time system is missed, no further activity is allowed and as such the utilisation of the processor should drop to 0 [2, p. 6-8].

Fixed-priority preemptive scheduling is employed for the AMS program. This implies that the priority of tasks are chosen before running the program and don't change during program execution. This means that a schedule for all tasks is generated before the execution of tasks, which constrains the possible behaviours of the system. The scheduling being preemptive implies that tasks may be preempted by other tasks and/or interrupt service routines (ISRs) which have a higher priority [2, p. 155]. Tasks are therefore structured such that they can be preempted with no side effects.

Another property of the system is that all tasks are independent (no task waits for information from another task before executing) [2, p. 156]. Furthermore all tasks are periodic with deadlines equalling their period times, and all tasks have the same period time in the case of the AMS. This allows the scheduler to employ rate monotonic scheduling [2, p. 209]. Since all tasks have the same periodicity other properties of the tasks are used to motivate the chosen priorities of the systems tasks.

The RTOS used for the program is FreeRTOS. FreeRTOS is a simple RTOS kernel for embedded systems distributed under the open-source MIT license, and is written in C [10].

B. Queues

Queues can enable mutually exclusive shared resources for inter-task communication. Mutual exclusion enables task independence [11]. Queues are first in first out (FIFO) type data buffers with set maximum queue lengths.

There are four types of operations possible: Sending, in which a task enters data in the back of the queue. If the queue is full it will tell the sender that it failed, and then discard the data. Receiving, in which a task takes a piece of data from the front of the queue. If the queue is empty, it will tell the receiver that it failed and return nothing. Peeking, which is the same as the receive operation except the data is left in the queue. If the queue length is one, there is a operation called overwrite in which the currently held data is overwritten. Since data is copied (not passed) when executing any operation, preemption of the data receiving task is assured since the copied data cannot be altered once it has been taken.

C. Schedulability

As follows are schedulability criteria for a system using a rate monotonic scheme [2, p. 162]. One criterion is, given N tasks of periodicity T with worst case execution time C , have the system satisfy the following relation:

$$\sum_{k=1}^N \frac{C_k}{T_k} = U \leq N(2^{(1/N)} - 1) \quad (1)$$

Where U is the total utilisation. Another sufficient criterion is that the worst case response time of every task should be lower than it's deadline. The worst case response time R can be calculated with the recursive relation (7.7) in [2, p. 168].

D. Multilayered Mealy Machine

State machines may be used to implement sequential and temporal logic using a periodic task, such as the various precharge procedures mentioned in the problem formulation. One paradigm is the multilayered mealy machine as described in this section. A complex system containing groups of states that all have a common exit condition can be represented in multiple layers, by placing state machines inside of individual states of other state machines.

The entry point to the state machine is denoted 0. State transitions are formulated as $[condition]\{statement\}$ where the conditions and statements are written as pseudo code. If the condition is fulfilled, the statement is executed and the state machine progresses to a new state. Every state implicitly has a hidden and empty transition onto itself that it will utilise when there are no other fulfilled transitions. An empty condition is always fulfilled, and an empty statement does nothing.

Every consecutive time step in which a state reenters itself (from itself), the inner state machine may iterate one step. If a state is reentered from another state, it's inner state machine is started from 0. Flags may be raised within any state which triggers an exit condition from the state. An exhaustive example is found on Figure 7.

The paradigm closely resembles MathWork's Stateflow [12], but is comparatively restricted in it's functionality.

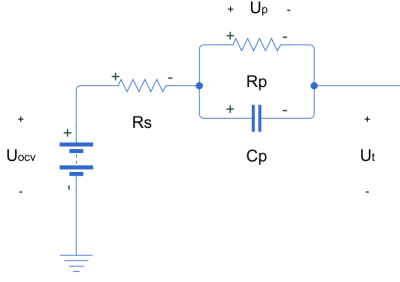


Fig. 4. A simple model of a cell.

E. Equivalent Circuit Model

Figure 4 depicts an equivalent circuit model which has been employed for the AMS cells. The measurable terminal voltage U_t and the inferred cell polarisation voltage U_p has the following state space equations:

$$\dot{U}_p = -\frac{1}{\tau_p}U_p + \frac{1}{C_p}I \quad (2)$$

$$U_t = U_{ocv} - U_p - R_s I \quad (3)$$

Where I is the current passing through the cell, R_s is the equivalent series resistance, R_p is the equivalent polarisation resistance and C_p is the equivalent polarisation capacitance (making $\tau_p = R_p C_p$ the first order time constant of the model). U_{ocv} is the voltage level across the actual cell. Discretising this model given a constant system sample time of T_s yields;

$$U_{p,k} = \exp\left(-\frac{T_s}{\tau_p}\right)U_{p,k-1} + (1 - \exp\left(-\frac{T_s}{\tau_p}\right))R_p I_{k-1} \quad (4)$$

$$U_{t,k} = U_{ocv} - U_{p,k} - R_s I_k \quad (5)$$

Two suitable variables to add onto the state space equations are the contained charge Q_k and total capacity $Q_{tot,k}$. They have the following equations:

$$Q_k = Q_{k-1} + T_s I_k \quad (6)$$

$$Q_{tot,k} = Q_{tot,k-1} \quad (7)$$

The change in $Q_{tot,k}$ is very small over a given timestep, so it will only progress due to the stochastic noise introduced by the modelled unscented kalman filter. This model is largely based on prior work in [13] with modifications inspired by the multi-timescale extended kalman filter in [14, p. 120].

The parameters R_s , R_p , τ_p and U_{ocv} are taken to be functions of the state of charge (SOC) of the cell. SOC being the quotient of current charge and total capacity is defined as $z_k = Q_k/Q_{tot,k}$. Tables tabulating $R_s(z)$, $R_p(z)$, $\tau_p(z)$ and $U_{ocv}(z)$ were produced in [13].

State of health (SOH) is defined here as being the quotient of total capacity to initial total capacity $q_k = Q_{tot,k}/Q_{tot,0}$.

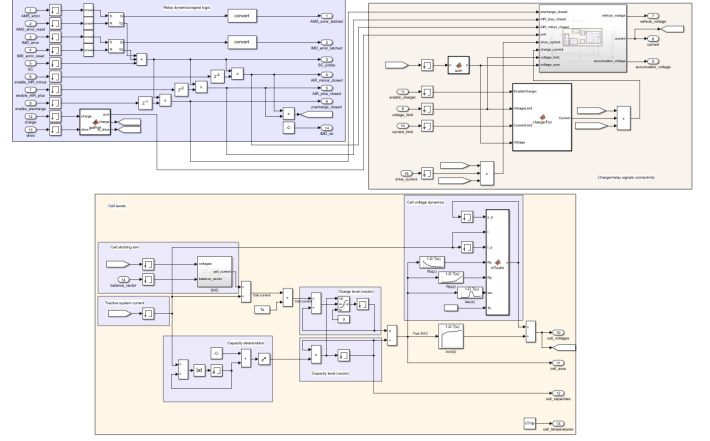


Fig. 5. The inside of SIM0. The blue block includes relay dynamics (delayed/failed relay toggling). The brown block includes a matlab function which imitates how the real charger works, as well as the circuit model in 2. The yellow block includes the cell engine and the cell balancing mechanic.

V. METHODOLOGY

A. Simulation and code generation

In order to reliably implement all of the tasked features, simplified models for the inverter, the charger and BECKY including all 252 cells was created by the authors. The system may therefore be simulated using MathWork's Simulink [15]. These models are placed in a subsystem called SIM0 5. SIM0 produces a host of outputs including all cell voltage levels, all true state of charges, capacities (with deterioration), relay states with relay switching dynamics (delays/failures), accumulator current, accumulator voltage as well as the vehicle voltage.

The mean cell voltage and the accumulator current is input into the cell state estimator CSE which is loosely based on state estimators such as the one freely available in [16]. It is depicted in 6, along with SIM0.

SIM0 interplays with a Stateflow state machine in 6 which implements the logic required for driving, charging and balancing the simulated cells.

The state machine is reconstructed using the framework established in the multilayered mealy machine section. Functions, variables and callbacks are generated using a Python script, which implements the state machine. The machine can therefore be re-structured and re-generated if alterations need to be made.

Using the Simulink embedded coder [17] configured for the Cortex-M4 architecture all the appropriate *.c* and *.h* files needed to build SIM0 and CSE are generated and built on the MC.

B. Program Description

To make the system easily analysed and easily implemented, fixed priority preemptive scheduling is employed. Further-

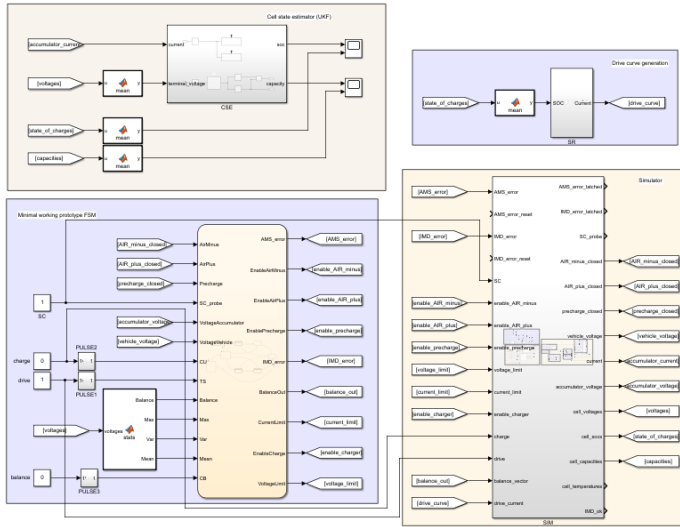


Fig. 6. These are the various elements used to verify the system components. In the leftmost blue block there is the minimal working prototype for the state machine. The yellow block includes SIM0. The rightmost brown block includes the CSE. There is an additional SR-block which generates a current triangle wave.

more, the tasks will be fully independent of each other and all signals are transmitted via a number of single element queues;

- IVT_accumulator_voltage; the accumulator voltage measured by the IVT,
- IVT_accumulator_current; the accumulator current measured by the IVT,
- IVT_vehicle_voltage; the vehicle side voltage measured by the IVT,
- accumulator_current; the accumulator current again, but is always emptied after it has been read,
- DBU_drive; a signal to from the vehicle's dashboard unit to start driving,
- DBU_balance; a signal to from the vehicle's dashboard unit to start balancing,
- CU_charge; a signal to from the vehicle's charger unit to start charging,
- CU_balance; a signal to from the vehicle's charger unit to start balancing,
- cell_voltages; the gathered cell voltages,
- cell_temperatures; the gathered cell temperatures,
- temperatures; the auxiliary temperatures,
- error_code; the error code (if any)

The interplay of the tasks and single element queues including which queue operations are used by what tasks is illustrated by Figure 8. This use of queues implies that no task ever waits for another task to finish (i.e. they are independent). But this also means that if the flow of critical data is interrupted for any reason the program will detect this (since it will try to read from an empty queue), raise an error, and enter an error state.

The deadlines of the program tasks named in the program structure section are all set to be 100 ms, and the period time of the tasks are set to be 100 ms as well.

C. Program structure

The program receives data from four different interrupt services routines and one data gathering task which itself receives data from previously executed tasks. The data is put into queues and later accessed by four data processing tasks. If any error occurs, or if any error is raised, a corresponding error code will be broadcast on CAN. The error signal will subsequently be lowered by an error task if the error doesn't persist. If any task causes the program to halt, or causes any task to exceed it's own deadline, an independent watchdog timer will reset the MC.

The following tasks may raise an AMS error or an IMD error signal. This shuts down the TS as described in section II-B4. The TS being shut down corresponds to entering a safe state.

1) *Temperature monitor ISR*: An ADC on the MC is set up to sample at a constant frequency, and set up with direct memory access so that a buffer in memory is continuously filled with 128 interleaving ADC-readings from 4 different thermistor circuits. When half of the conversions are done, the corresponding half of the buffer is averaged and the temperatures are calculated.

2) *Signal monitor ISR*: A digital IMD signal is connected as an external interrupt with a falling edge trigger. If this interrupt is triggered, an IMD error is immediately raised as this indicates that insulation has been compromised.

The system anticipates a propagation delay from the setting of relay states and the actual detectable switching of relay states. Therefore the relay states readings are set as external interrupts with falling and raising edge triggers. As soon as the relays switch states, they are compared with the values which are set by the MC. If the state of the relay differs from their intended state an AMS error is raised.

3) *CAN RX ISR*: When CAN messages are received by the MC, they are input into a single element queue as to guarantee to any consuming task that they are receiving an up to date value. There are 3 periodically received messages from the IVT, namely the vehicle voltage (the voltage over the inverter), the accumulator current and the accumulator voltage. It is vital that any task which requires these, but do not receive them, raise an AMS error. There are also a couple of sporadic CAN messages that tell the program to enter the drive, charge or balance modes. Single element queues are used to guarantee that new values are being received every time the SM task or the CEM task is being called (by having those task use the receive operation).

4) *IMD ISR*: A timer on the MC is configured to receive and interpret a PWM signal which is generated by the IMD. The timer will periodically trigger an ISR, which yields the duty cycle and frequency of the received signal. The signal encodes the level of insulation of the AIRs in relation to

the chassi. If the recorded insulation level is lower than 500 Ohm/volt the system should immediately raise an AMS error, as this could potentially prelude a short circuit. The IMD's IMD_ok signal is directly connected to the IMS error signal. The IMD ISR is therefore mainly used to prove that the IMD works as intended.

5) *Cell monitoring task*: This task may not launch until the IVT accumulator current has been received one time. Once active it will attempt to receive the IVT accumulator current every period. If it fails to receive from the corresponding queue, it will launch an AMS error. It is therefore necessary that the deadline of this task (100 ms) is greater than the periodicity of the IVT messages (60 ms [18]). The task then gathers all 126 voltages and 60 temperatures from the AMS slaves, one by one, and scrutinises all data points. The STM32F407VG SPI hardware routine determines if the data is corrupted. If it is, an error is immediately raised and the state machine enters the error state described in 7.

The critical bounds that apply to the voltage levels, temperature levels and current level were determined from a data sheet found in [4]. The lower and upper cell temperature bounds are -19 °C and 59 °C respectively. The lower and upper cell voltage bounds are 2.81 V and 4.19 V respectively. The lower and upper parallel cell current bounds are -20 A and 40 A respectively. These are used in the level-time constraint algorithm in section V-D.

Following this scrutiny, the task will execute the level-time constraint algorithm 1 on the cell voltages, cell temperatures and the accumulator current in compliance with rule 2. If all tests are passed, this task will send all of the scrutinised data on the appropriate data queues.

6) *State machine task*: This task executes the state machine in Figure 7, which is developed in section V-A. The purpose of this task is to execute all of the necessary behaviour for the driving, charging and balancing modes.

The inputs of the state machine are:

- accumulator_voltage; the voltage over the battery,
- accumulator_current; the current draw from the battery,
- vehicle_voltage; the voltage over the inverter or charger,
- air_minus_closed; the state of the negative side AIR,
- air_plus_closed; the state of the positive side AIR,
- precharge_closed; the state of the precharge relay,
- minimum_cell_voltage; the minimum cell voltage,
- maximum_cell_voltage; the maximum cell voltages,
- cell_voltages_variance; the variance of the cell voltages,
- SC; a boolean which is high if the relay voltage source is high,
- imd_error; the state of the imd error latch,
- ams_error; the state of the ams error latch,
- balance; the signal which initiates the balance mode,
- drive; the signal which initiates the drive mode,
- charge; the signal which initiates the charge mode,
- charger_is_awake; a boolean indicating whether the charger is on.

The outputs of the state machine are:

- close_air_plus; a signal which sets the state of the positive side AIR,
- close_air_minus; a signal which sets the state of the negative side AIR,
- close_precharge; a signal which sets the state of the precharge relay,
- enable_charger; a signal which sends a CAN message telling the charger to start charging.
- error; a signal which raises an AMS error.

7) *Cell state estimator task*: This task receives the cell voltages and the accumulator current from the queues that were sent from the cell monitoring task. It uses these to progress the unscented kalman filter which is generated using Simulink in section V-A using the model proposed in section VII-F.

8) *Cooling task*: This task receives the cell temperatures from the queues that were sent from the cell monitoring task. It hosts a PID controller which is set to proportional gain and sets the duty cycle of a fan which cools the battery. It uses the maximum cell temperature as input, and 20 degrees celsius as a reference. The gain is such that the duty cycle is 100% at 40 degrees celsius.

9) *CAN TX Scheduler task*: This task receives the cell temperatures, cell voltages, the vehicle/accumulator voltages and error code. It also gets the states of various system signals such as the relay states. It broadcasts these on CAN in compliance with rule 3.

10) *Error handler task*: This task will receive the error code which was sent any time an error was raised. Since raising an error corresponds to setting a signal in the physical circuit, this task will reset the error setting signal after 500 ms if no further error is detected/received, so that the error signal may be reset by an electrical safety officer at a later time.

11) *Independent watchdog task*: The MC's independent watchdog (IWDG) is configured such that if any program unexpectedly exceeds its worst case response time, the MC is reset (powered-down and then powered-up). The IWDG is set to count down a timer independently of any activity on the processor, and upon reaching 0 it performs a reset. The purpose of the task is to rewind/reset the timer as to prevent the system reset from occurring in the absence of any program errors.

D. Level-Time Constraint Algorithm

Given an array \vec{v} of N values, let's say an error should be raised if the value has been faulty for T seconds. Assume also that \vec{v} is sampled with a constant sample period of T_s seconds. A reasonable assumption is then that if a value in \vec{v} has been faulty for more than $S_T = 1 + \lfloor T/T_s \rfloor$ consecutive samples, it can be considered as having persisted for more than T seconds.

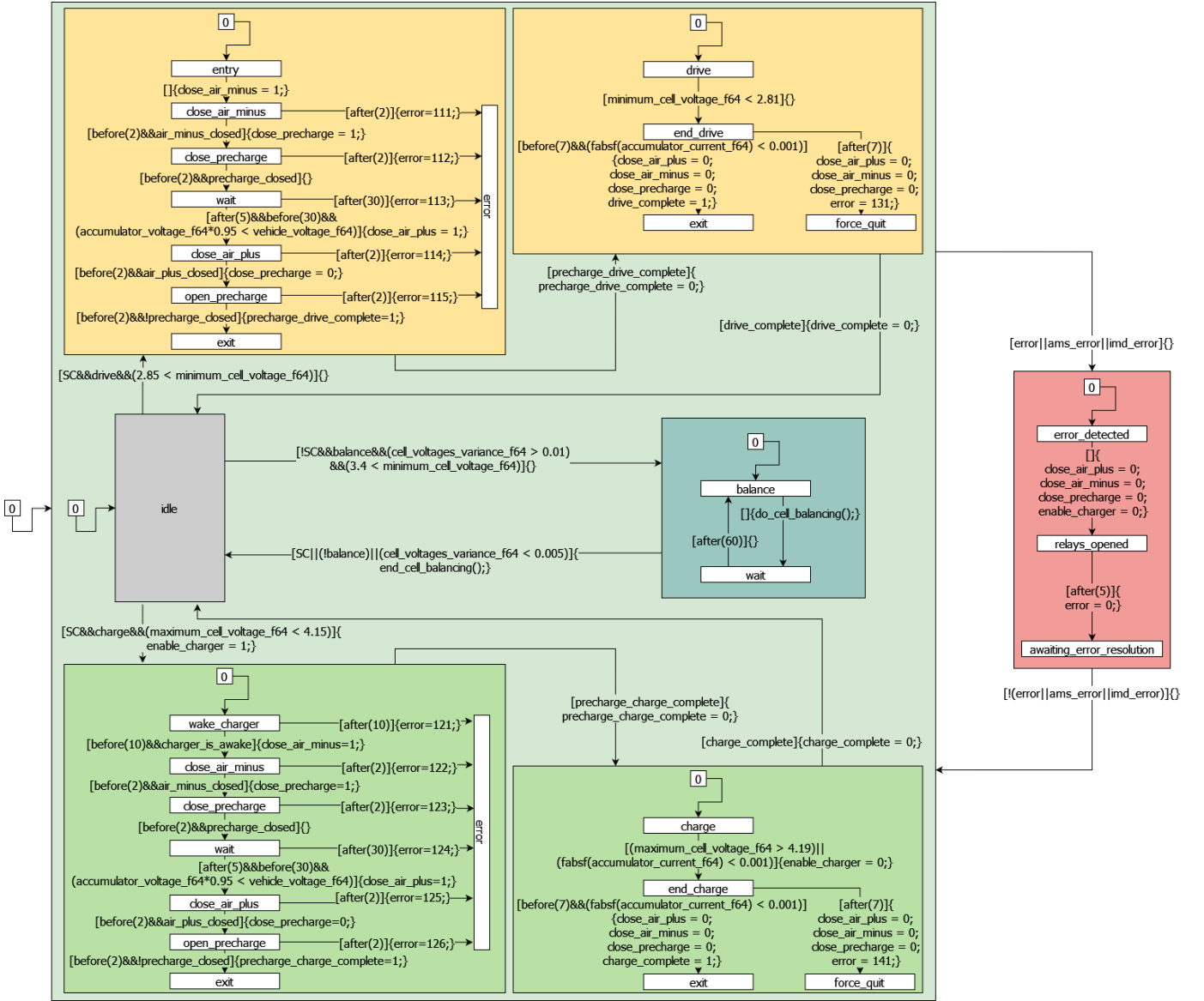


Fig. 7. The entire state machine. The light green state contains the main state machine, while the red state contains a state machine which handles errors by opening the relays. The yellow states contain the precharge drive and drive state machines. The green states contain the precharge charge and charge state machines. The marine state contains the balance state machine.

Algorithm 1 showcases how a vector \vec{v} whose values have a common critical upper boundary U and critical lower boundary L can be evaluated in regards to a time-level violation; The sample constraint S_T is calculated from the system sample time T and the time constraint T . The code loops over every vector index j , and if it finds that a value \vec{v}_j is lower than L , then the lower error vector \vec{l}_j is incremented. If it finds that a value \vec{v}_j is higher than U , then the upper error vector \vec{u}_j is incremented. The the number of error vector increments exceeds the sample constraint for either error vector, the program is returns an integer corresponding to the fault. If there is no fault, an integer corresponding to no fault is returned.

E. Cell Balancing Strategy

Once a balance signal for a cell is set, it's terminals are connected with a resistance of R for a set time T . The goal of the cell balancing strategy is to decrease the variance in voltage of all cells. This is done by drawing current out of a quarter of all cells, specifically those who have the highest voltage. The voltage is monotonic to the charge stored in the cell [4], so the voltage will decrease when subject to the current draw.

In algorithm 2 the voltage vector \vec{U} of length N is balanced by setting the balance/short circuit control vector \vec{B} high at the appropriate indices (the i :th entry of \vec{B} draws current out of the i :th cell), and waiting T seconds before resetting the balancing vector and calculating the variance in voltage σ_U^2 . This is repeated until a tolerance ϵ exceeds the calculated σ_U^2 .

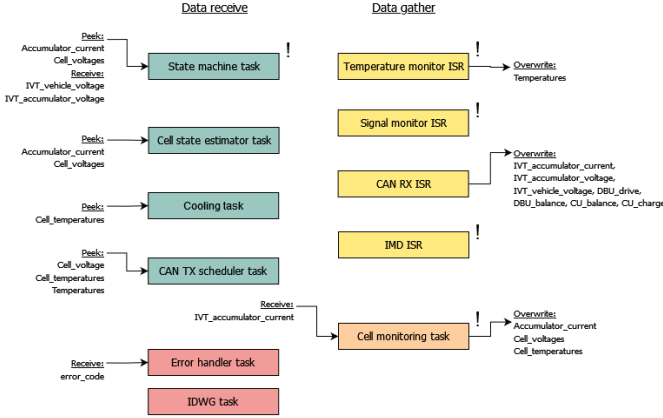


Fig. 8. The data flow of the program. Exclamation points indicate the capacity of a task to raise a program-halting error. The blue boxes are receiving tasks that after receiving data from queues for the first time, will launch an error if they do not receive any further data. Red boxes are receiving tasks that respond to various fault conditions. The peach box is a task which creates program data after receiving the latest data from an ISR. The yellow boxes are ISRs that create data.

Algorithm 1 Algorithm for determining level-time constraint failure in sampled system

```

1: global variables
2:  $\vec{u} \leftarrow \mathbf{0}$ 
3:  $\vec{l} \leftarrow \mathbf{0}$ 
4: end global variables
5: procedure VALUEISCritical( $\vec{v}, T, T_s$ )
6:    $S_T \leftarrow 1 + \text{floor}(T/T_s)$ 
7:    $j \leftarrow 0$ 
8:   while  $j < N$  do
9:      $j \leftarrow j + 1$ 
10:    if  $\vec{v}_j < L$  then ▷ Under value guard
11:       $\vec{l}_j \leftarrow \vec{l}_j + 1$ 
12:      if  $S_T < \vec{l}_j$  then
13:        return 2 ▷ 2 indicates critically low value
14:      end if
15:    else
16:       $\vec{l}_j \leftarrow 0$  ▷ Reset if no under value
17:    end if
18:    if  $U < \vec{v}$  then ▷ Over value guard
19:       $\vec{u}_j \leftarrow \vec{u}_j + 1$ 
20:      if  $S_T < \vec{u}_j$  then
21:        return 1 ▷ 1 indicates critically high value
22:      end if
23:    else
24:       $\vec{u}_j \leftarrow 0$  ▷ Reset if no over value
25:    end if
26:  end while
27:  return 0 ▷ 0 indicates no error
28: end procedure

```

'sort_descending' is a function which sorts the vector \vec{I} such that $\vec{U}_{\vec{I}_{k-1}} \leq \vec{U}_{\vec{I}_k}$ for all indices k (except $k = 0$).

Algorithm 2 Algorithm for balancing cell levels

```

1: procedure STRATEGY( $\vec{U}$ )
2:    $\vec{B} \leftarrow [0, 1, \dots, N]$ 
3:    $\sigma_U^2 \leftarrow \text{Var}(\vec{U})$ 
4:   while  $\epsilon < \sigma_U^2$  do
5:      $\vec{I} = \text{sort\_descending}(\vec{I}, \vec{U})$ 
6:      $k \leftarrow 0$ 
7:     for  $k < N/4$  do
8:        $\vec{B}_{\vec{I}_k} \leftarrow 1$ 
9:     end for
10:    Wait for  $T$  s
11:     $j \leftarrow 0$ 
12:    for  $j < N$  do
13:       $\vec{B}_j \leftarrow 0$ 
14:    end for
15:     $\sigma_U^2 \leftarrow \text{Var}(U)$ 
16:  end while
17: end procedure

```

F. SOC/SOH Estimation

The discrete-time state space equations of the equivalent circuit model in section IV-E are written as follows;

$$x_{k+1} = f(x_k, u_k) + \omega_k \approx A_k x_k + B_k u_k + \omega_k \quad (8)$$

$$y_k = h(x_k, u_k) + \nu_k \quad (9)$$

Where f is the state transition model and A_k, B_k make up a linear approximation of the state transition model f with process noise ω_k at time step k . h is the observation model of the equivalent circuit model with observation noise ν_k . The model is built using relations (4), (5) and (6);

$$x = [U_p, Q, Q_{tot}], \quad u = I, \quad y = U_t \quad (10)$$

$$A = \begin{bmatrix} \exp(-\frac{T_s}{\tau_p(z_k)}) & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (11)$$

$$B = \begin{bmatrix} (1 - \exp(-\frac{T_s}{\tau_p(z_k)}))R_p(z_k) \\ T_s \\ 0 \end{bmatrix} \quad (12)$$

$$y_k = U_{t,k} = U_{ocv}(z_k) - U_{p,k} - R_s(z_k)I_k \quad (13)$$

The desired outputs may then be constructed:

$$z_k = Q_k/Q_{tot,k}, \quad q_k = Q_{tot,k}/Q_{tot,0} \quad (14)$$

The state space equations are described in section IV-E and the generation of the code is described in V-A.

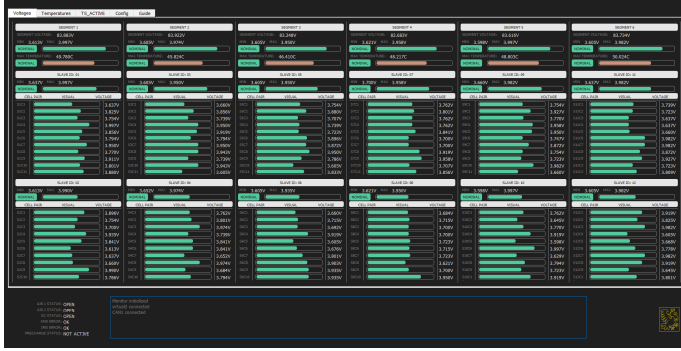


Fig. 9. All cell voltages being continuously displayed. At the bottom relay states are displayed.

G. GUI

In order to comply with EV 5.8.11 a GUI was programmed using the python modules canlib and PyQt5, developed by Kvaser [19] and The Qt Company [20] respectively. The computer receives CAN messages when connected to the vehicle's CAN system using a Kvaser CAN-USB connector. The GUI continuously displays the most recent cell voltages and cell temperatures, and is also capable of sending CAN messages to the AMS-master, which is used for testing purposes when the accumulator container is not in the car.

VI. TESTING SET-UP

In order to evaluate the performance of the program, a test set-up is necessary. The data from the testing is recorded using SEGGER Systemview [21] which is capable of streaming data (vehicle voltage level, all cell voltages, current state of the state machine, and error status) and events (task release time, task completion time) to the computer using a debugger. These are used to monitor successful initiation of the state machine sequences drive, charge and balance and are also used to test whether the concurrent monitoring of cell voltages, cell temperatures, accumulator current and insulation is rule compliant.

In order to test whether the system works in realistic settings other facets of the electric vehicle are emulated. The GUI in section V-G is used to send the drive-, charge- and balance initiating signals DBU_drive, DBU_balance, CU_drive, CU_balance in the stead of the respective vehicle subsystems. As can be seen in Figure 10 the accumulator was connected (using orange high voltage rated cables) to supply a high voltage level to BECKY. The vehicle side of the AIRs are connected to a capacitor bank (but no inverter) like in 2 in order to perform the precharge drive procedure. No inverter, charger or cell-balancing hardware is currently available for testing.

VII. RESULTS AND ANALYSIS

A. Proof of Schedulability

A Table of recorded tasks names, priorities p , period times T , worst case computation times (WCETs) C , task utilisation

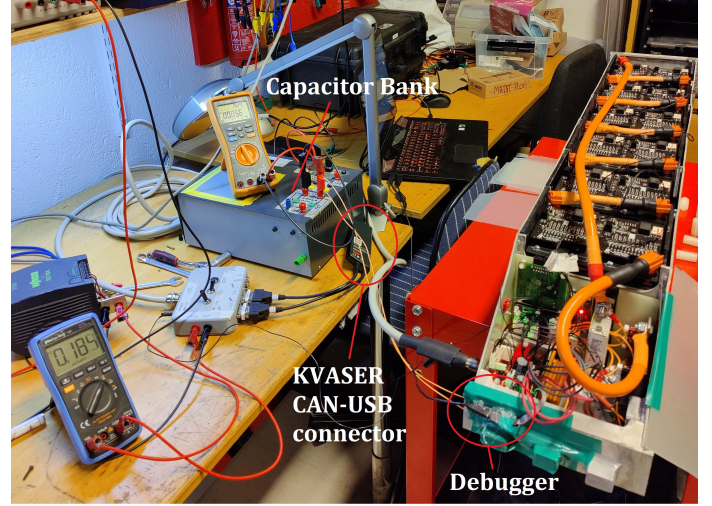


Fig. 10. The set-up used for testing the system in a realistic setting.

TABLE I
TABLE DETAILING THE BEHAVIOUR OF THE SYSTEMS TASKS

Task	p	T [ms]	C [ms]	U	ΣU	R [ms]
COOL	29	100.00	1.63	0.02	0.02	1.63
CSE	28	100.00	3.64	0.04	0.05	5.27
CAN	27	100.00	6.54	0.07	0.12	11.81
COM	26	100.00	52.18	0.52	0.64	63.99
SM	16	100.00	1.33	0.01	0.65	65.32
ERROR	9	100.00	0.38	0.00	0.66	65.70
IWDG	8	100.00	0.39	0.00	0.66	66.09
-height						

U , running sum of utilisation ΣU and the resulting worst case response times R can be found in Table I. The data is recorded on the realistic test set-up. The IVT data is being acquired on CAN, the cell voltages and cell temperatures are being transmitted on CAN, while concurrently being acquired from the AMS-slaves using isoSPI. The drive state is sporadically entered and exited, and various errors are purposefully being triggered to stress the system during the recording of the data. The WCETs C in the table have been increased by 20% in order to introduce a safety margin.

COOL, CSE, CAN, CEM, SM, ERROR and IWDG represents the cooling, cell state estimation, CAN TX Scheduler, cell monitor, state machine, error handler and independent watch dog tasks in sections V-C8, V-C7, V-C9, V-C5, V-C6, V-C10 and V-C11 respectively.

Using equation (1) with $N = 6$ one finds that the upper limit for program utilisation is 73%, which is well above the calculated 66% (including a safety margin of 20%) in Table I. Furthermore none of the response times of the tasks exceed their respective deadlines; the program is therefore fully schedulable.

B. Level-Time Constraint Compliance

Assuming that schedulability has been proven all tasks should successfully execute within their allotted deadline of 100 ms.

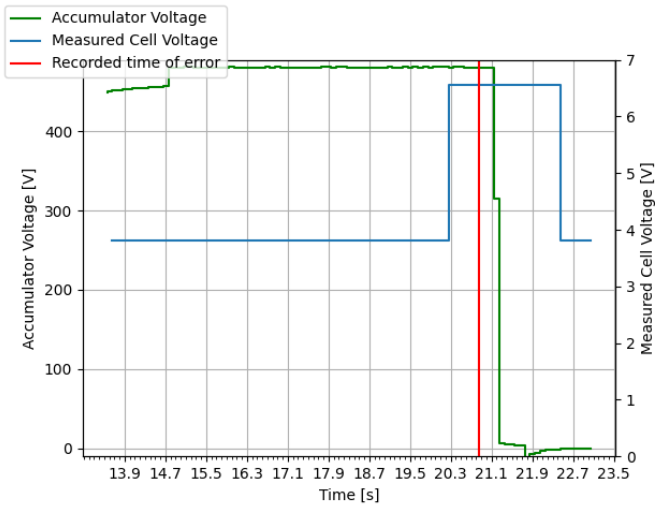


Fig. 11. A recorded instance of a cell over voltage resulting in an AMS error being triggered.

This means that the the cell monitoring task executes algorithm 1 virtually periodically. A recorded instance of over voltage triggering an AMS error per algorithm 1 after a successful precharge can be found in Figure 11. This error is followed by the subsequent shutdown of the TS. The AMS error occurs after 600 ms (corresponding to 7 consecutive faulty samples) per rule 2. There is an additional 250 ms delay before the AIRs are closed, due to a timed capacitor circuit (allowed for by EV 6.1.5 in [3]).

C. Driving

The designed state machine is perfectly able to enter the driving mode given the appropriate input. A recorded instance of a HV precharge being performed in a realistic setting is displayed in Figure 12. As soon as the precharge procedure is done (the inverter voltage is 95% of the accumulator voltage), the insulation is immediately compromised by short circuiting the negative side AIR to the chassi. This error is detected within 30s, and leads to the timely shutdown of the TS (whenever the accumulator voltage drops).

D. Charging

A recorded instance of the simulated accumulator attempting charging according to the state machine procedure in Figure 7 is graphed in Figure 13. At the beginning of the test the cells had uneven levels of charge, and as such the charging is cancelled when one of the cells is fully charged ahead of the other cells. As per the state machine, the charging is prematurely turned off as to not risk putting the fully charged cell into a critical over voltage state. Should one wish to reach a higher total voltage a cell balancing procedure can be performed before resuming with the charging again.

E. Balancing

A recorded instance of the simulated cell voltages being balanced using algorithm 2 is graphed in Figure 14. It shows

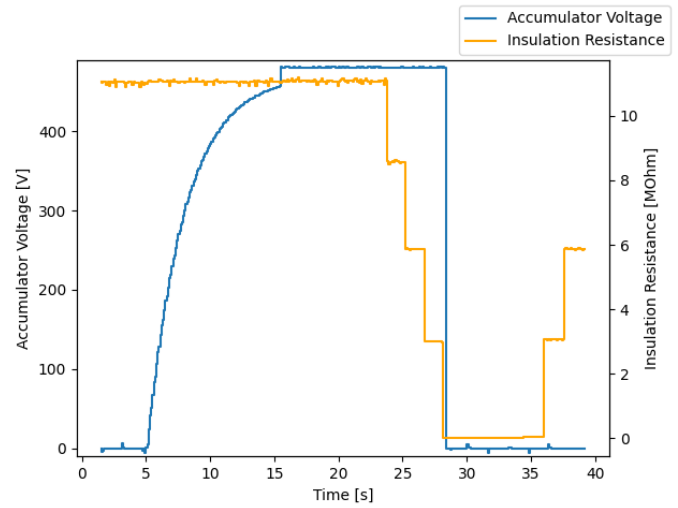


Fig. 12. A recorded instance of the AMS executing the precharge drive step and entering drive, and subsequently shutting down the TS due to a compromised insulation.

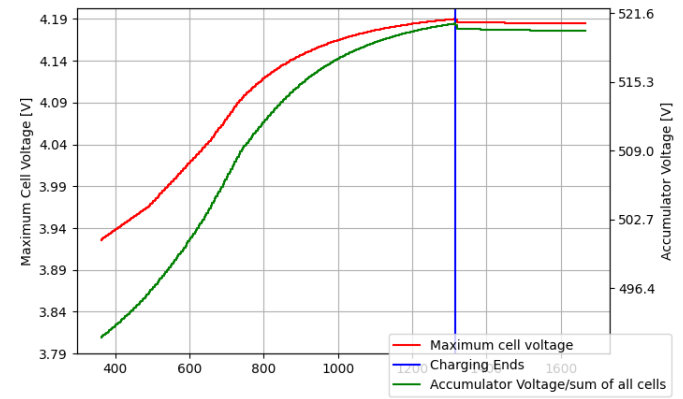


Fig. 13. A recorded instance of the simulated accumulator being charged.

that given time the cell voltages will reach the desired variance, and that the cell balancing operation may take several hours to complete. The voltage levels of 30 of the cells are displayed in the figure. The cells are shorted for 60 s at a time. They start out with an initial variance of 0.26 V^2 and end up at a variance of 0.0049 V^2 , where they meet the variance exit condition in the state machine in Figure 7. The cell voltages are organised into bands to show how different groups of voltage levels are affected by the procedure.

F. SOC/SOH estimation

A recorded instance of the simulated accumulator being driven using a triangle wave curve is recorded in Figure 15, where the estimated mean SOC is compared to the true SOC. The final estimated SOC is 0.639, the final true SOC is 0.627, and the final difference is 1.87%. In Figure 16 the estimated capacity deterioration is compared to the true capacity deterioration during the same drive. The true SOH is 96%, while the estimated SOH is 83%, a noticeable discrepancy.

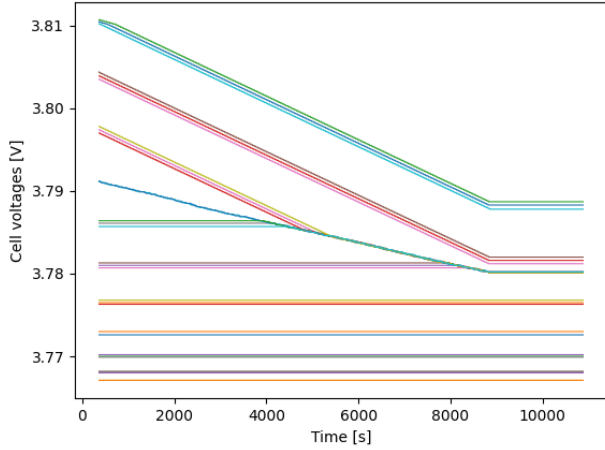


Fig. 14. A recorded instance of cell balancing over the course of 12000 s.

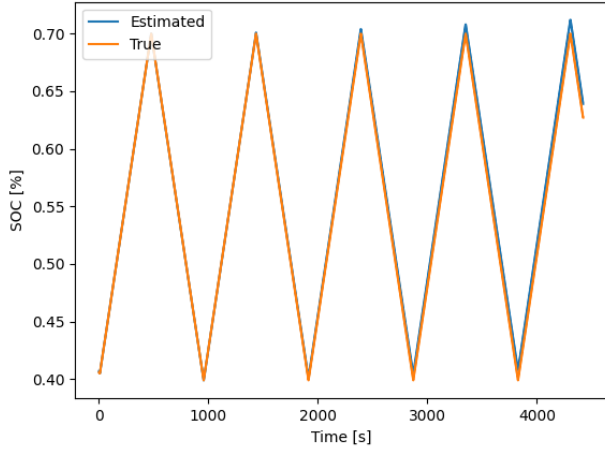


Fig. 15. The simulated AMS executing the precharge drive step and entering drive. It is then subject to a triangle wave current.

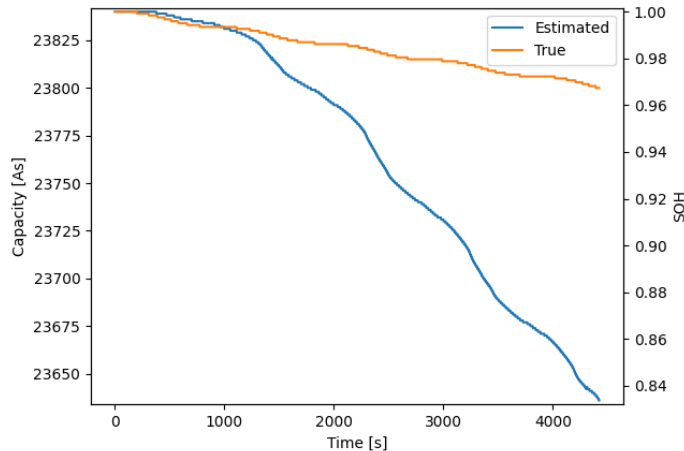


Fig. 16. This figure illustrates the estimated and true capacity/SOH deterioration in figure 15.

G. Rule compliance

In the following list the rule compliance is evaluated:

- FSG Rule 1, is proven in the result sections VII-C.
- FSG Rule 2, is proven in the result section VII-B.
- FSG Rule 3, is proven in section V-G.
- FSG Rule 4, is proven in section VII-C.
- FSG Rule 5, by design the charger is enabled and disabled as part of the state machine in Figure 7, discussed in section V-A and proven in VII-C.
- Rule 1, is shown in the result section VII-C.
- Rule 2, is shown in the result section VII-D.
- Rule 3, is shown in the result section VII-E.
- Rules 4, 5, 6, and 7 are all shown in section IV-C.

VIII. DISCUSSION

A. System Credibility

A point of discussion might be whether the simulated results translate into real world results. Since the drive state has been proven to work (it hasn't been tried in any substantial discharge test, but such a test poses new challenges to the underlying hardware) the bridging of the gap between simulated and real results has been demonstrated. It is therefore plausible to translate the simulated charging and balancing to real charging and balancing with minimal effort as well.

The system has run in real world scenarios for extended periods (more than 1 hour) and displayed accurate voltages and temperatures on the GUI for the whole period. This indicates that the system is stable and works as intended under non strained conditions. Every possible state machine sequence was not attempted during the recording of events in section VII-A since there is no hardware support charging and cell balancing (charging would fail and balancing would not do anything). Every state is composed of very little logic, so no new challenges would be poised by including them in the executed sequences. Furthermore, because the real-life system state (including relay states, voltage levels) is reset upon exiting the drive, charge and balancing state (whether exit was due to an error or not) there are no apparent factors that could affect the feasibility of the proof of schedulability.

The simulation framework in section V-A is exhaustive but could have more features; as can be seen in the Figure 14, noise has not been simulated. This would be a good addition to the model as it would showcase whether the algorithms and the state machine have an adequate level of noise immunity. Thermal factors such as cell heat generation and cell parameter heat dependency would also be appropriate to factor in. These factors can only be accurately simulated after having collected data on the batteries thermal properties in it's proper casing, which could not occur in the planned project timeline. This was therefore not considered for this project, but should be considered in future work.

SOC and SOH estimation are fairly credible. The procured SOC results in Figure 15 are very accurate, and the SOH

results in Figure 16 are fine. The real life performance of the estimation is only as accurate as the tabulated parameters in [13], whose accuracy can not be proven in the simulated environment. Assuming that the parameters are accurate, the simple estimation presented in this paper is adequate. A further inquiry into the estimation algorithms is deemed appropriate and this paper should serve as a starting point for future work.

Alterations to the product are inevitable. Chargers, inverters, cells may be replaced and need to be accommodated for. FSG rules are also subject to change and as such the rule compliance of the current system might be contested in the future.

B. System Adaptability

The system is overall very tunable. The state machine can be edited and re-generated. Factors such as level-time constraint boundaries and system sample times can be changed with predictable outcomes. If changes are made the schedulability can be proven again using the procedures outlined in this paper. If the hardware platform is not available for trials, the simulator may be used to evaluate future changes.

C. Future work

Before anything else the physical platform should be finished and the software should be further tested and proven. This implies incorporating the charger, the inverter and the cell-balancing hardware into the test set-up. And executing the proposed charging and balancing logic.

The SOC and SOH algorithms were rudimentary, and operated only on a mean aggregate of all cells. In the future one might want to calculate the SOC and SOH for every individual cell. This way one could supplement the voltage constraints with SOC/OCV constraints, which would increase the drive cycle bounds without compromising safety. This is because the critical voltage level bounds actually pertain to the OCV of the cell, but it is the terminal voltage level which is being monitored by the system. This often over/under estimates the measured voltage since the accumulator current changes the measured terminal voltage level per the equivalent circuit models series resistance.

If 126 SOC/SOHs are to be calculated (corresponding to every measurable cell voltage level), the CSE task would in theory then require a 126 times greater computation time (382 ms). One could simply increase the deadline of this task, while maintaining that rule 2 is still being complied to.

Another measure is to create a solution for streaming the various SOC, SOH and model parameter values to a computer as to preserve a historical record/database of the cell health. This would enable data analysis to be done in the future, and it would yield a more complete picture of the battery's performance and health. The cell model parameters tends to drift over time/with use, so one should either do careful

data analysis or simply re-tabulate the cell parameters per the procedure outlined in the 2021 vdpc report [13] routinely.

IX. CONCLUSION

The AMS software is fully rule compliant, though all of it's functionality has yet not been explored; charging and balancing remains to be tested in a realistic setting, and smarter SOC/SOH algorithms may be incorporated to increase the safety and performance of the system in the future. Since transitioning from simulated results to actual real life results was very successful for the proposed drive mode, doing the same for charging and balancing appears to be easily feasible. Due to the credibility and adaptiveness of the system, it will serve as a good platform for more experimental SOC/SOH algorithm development. The simulation suite and Simulink models can be used for planning and designing such algorithms before deploying the them on the actual system. If any changes need to be done to the state machine itself, the state machine is simply be re-structured and re-generated.

ACKNOWLEDGMENT

The authors would like to thank Matthias Becker for agreeing to supervise to project, as well as affording ideas and solutions for debugging and development of various facets of the system. The authors would also like to thank the KTH formula student team for proposing the project and Anita Kullen for making it possible to incorporate into course work.

REFERENCES

- [1] X. Wu and Song, "Safety issues in lithium ion batteries: Materials and cell design," *Frontiers in Energy Research*, vol. 7, Sep. 2019. [Online]. Available: <https://www.frontiersin.org/article/10.3389/fenrg.2019.00065>
- [2] K. Erciyes, *Distributed Real-Time Systems Theory and Practice*, 1st ed., ser. Computer Communications and Networks, Jul. 2019.
- [3] Formula Student Germany. (2021, Mar.) Formula student rules 2022, version: 1.0, rev-9c22985. Stockholm. [Online]. Available: https://www.formulastudent.de/fileadmin/user_upload/all/2022/rules/FS-Rules_2022_v1.0.pdf
- [4] Shenzhen Melasta Battery Co., Ltd. (2021, May) Lithium polymer (li-po) batteries/cells. Shenzhen Melasta Battery Co., Longhua, Shenzhen. [Online]. Available: <https://www.melasta.com/cells/lithium-polymer-li-po-batteries-cells-.html>
- [5] Analog Devices, Inc. (2018, Jun.) Ltc6804-1. Analog Devices, Inc., Wilmington, Massachusetts. [Online]. Available: <https://www.analog.com/en/products/Ltc6804-1.html>
- [6] STMicroelectronics. (2022, May) High-performance foundation line, arm cortex-m4 core. Plan-Les-Ouates. [Online]. Available: <https://www.st.com/en/microcontrollers-microprocessors/stm32f407vg.html>
- [7] Analog Devices, Inc. (2018, Jun.) Ltc6820. Analog Devices, Inc., Wilmington, Massachusetts. [Online]. Available: <https://www.analog.com/en/products/Ltc6820.html>
- [8] Current Ways, Inc. (2017, Jun) 3kw air-cooled ev battery charger (225-450vdc). Current Ways, Buena Vista, California. [Online]. Available: <https://currentways.com/ev-battery-chargers/3k-112-450vdc/3kw-bc-series-air-cooled-ev-battery-charger-225-450vdc/>
- [9] C. Walls, *Embedded RTOS design : insights and implementation*. Cambridge, Massachusetts: Elsevie, Dec 2020.
- [10] Real Time Engineers Ltd. (2021, Mar) Freertos. Real Time Engineers Ltd., Bristol, North Somerset. [Online]. Available: <https://www.freertos.org/index.html>
- [11] Arm Ltd. (2021, Oct) Message queue. Arm Ltd., Cambridge. [Online]. Available: https://www.keil.com/pack/doc/CMSIS/RTOS2/html/group__CMSIS__RTOS__Message.html

- [12] The MathWorks, Inc. (2022, May) Represent event-driven systems in simplified graphic form. The MathWorks, Inc., Natick, Massachusetts. [Online]. Available: <https://se.mathworks.com/discovery/state-machine.html>
- [13] KTH Formula Student. (2021, Jan.) Kthfs vdpc report 2021. KTH Formula Student, Stockholm.
- [14] R. Xiong, *Battery Management Algorithm for Electric Vehicles*, 1st ed. Beijing: Springer, Oct 2020.
- [15] The MathWorks, Inc. (2022, May) Simulink is for model-based design. The MathWorks, Inc., Natick, Massachusetts. [Online]. Available: <https://se.mathworks.com/products/simulink.html>
- [16] —. (2022, May) Nonlinear state estimation of a degrading battery system. The MathWorks, Inc., Natick, Massachusetts. [Online]. Available: <https://se.mathworks.com/help/control/ug/nonlinear-state-estimation-of-a-degrading-battery-system.html>
- [17] —. (2022, May) Embedded coder. The MathWorks, Inc., Natick, Massachusetts. [Online]. Available: <https://se.mathworks.com/products/embedded-coder.html>
- [18] Isabellenhütte Heusler GmbH Co. KG. (2018, Apr.) Ivt-s series for battery management systems. Isabellenhütte Heusler GmbH Co., Dillenburg. [Online]. Available: <https://www.isabellenhuette.de/en/precision-measurement/standard-products/ivt-s-series>
- [19] Kvaser AB. (2022, May) Kvaser, can. Kvaser AB, Mölndal. [Online]. Available: <https://www.kvaser.com/>
- [20] The QT Company. (2022, May) Qt. The QT Company, Espoo. [Online]. Available: <https://www.qt.io/>
- [21] SEGGER Microcontroller GmbH. (2022, May) Systemview — analyzing embedded systems. SEGGER Microcontroller GmbH, Monheim am Rhein. [Online]. Available: <https://www.segger.com/products/development-tools/systemview/>

Prototype of a Charge Controller for a Formula Student Electric Vehicle

Yves Obreykov and Fredrik Stoltz

Abstract—The demand for electric vehicles is ever-increasing and as such, there needs to be an efficient and easy way to charge their batteries. Aiming to simplify the use of chargers this report has tackled the challenge of developing a prototype for charging a Formula Student Electric Vehicle. This prototype was named Charge Controller and is what a user will interact with when charging the vehicle's battery. In this project, a new architecture for the charging process has been designed for both hardware and software. Initial tests prove that the charging of the vehicle can be done in a simple manner. The hardware has been designed, produced, partially assembled and partially tested. The software has been tested using a development board which demonstrates that the design works.

Sammanfattning—Efterfrågan på elfordon ökar ständigt och därför måste det finnas ett effektivt och enkelt sätt att ladda batterierna. För att förenkla användningen av laddare har man i denna rapport tagit sig an utmaningen att utveckla en prototyp för att ladda ett elfordon från Formula Student. Denna prototyp fick namnet Charge Controller och är det som användaren kommer att interagera med när fordonets batteri laddas. I detta projekt har en ny arkitektur för laddningsprocessen utformats för både hårdvara och mjukvara. De första testerna visar att laddningen av fordonet kan göras på ett enkelt sätt. Hårdvaran har konstruerats, tillverkats, delvis monterats och delvis testats. Programvaran har testats med hjälp av ett utvecklingskort, vilket visar att designen fungerar.

Index Terms—PCB, CAN, Charging, LVGL, embedded GUI, HMI.

Supervisor: Carl-Mikael Zetterling

TRITA number: TRITA-EECS-EX-2022:136

I. INTRODUCTION

A. Background

KTH Formula Student (KTHFS) is a student driven project where students from KTH produce an electric vehicle and compete in engineering competitions such as Formula Student Germany against other teams from all around the world [1]. The vehicle is driven by a high voltage battery.

The charging process of the battery is currently managed manually with a laptop which is a time consuming process and means that it is not possible to automate the charging cycle. Additionally, it is difficult to read real-time metrics of the battery during charging. Since high voltages are involved in the process it is also a potentially dangerous process. It would therefore be beneficial if the process would be simplified with higher reliability and predictability to minimize the risks.

B. Project formulation

The aim of this thesis project is to simplify the charging task by developing both hardware and software for a prototype that can automate the charging cycle of the battery. In the future, this will also serve as a platform for developing an easy-to-use interface for reading real-time charging metrics such as state of charge, state of health and monitoring the battery temperature. The prototype will be referred to as the Charge Controller (CC) in this report and will be mounted on the square gray box seen in Fig. 1 where it is also possible to see the two chargers at the bottom of the wagon.

The hardware part of the project consists of designing and constructing a Printed Circuit Board (PCB) with a touchscreen as a Human-machine interface (HMI). The HMI will allow the user to start and stop the charging process as well as monitoring charging parameters. The PCB will also incorporate hardware for adding functionality to the CC in the future, such as thermistors and buttons.

The software part of the project consists of programming a microcontroller with the necessary GUI functions that will interact with the user, and also handle all the communication with the battery. The communication between the different systems will be done using the communication protocol Controller Area Network (CAN) [2].

Many of the building blocks for this project will utilize various solutions from previous projects done within KTHFS. This is mainly in regards to the hardware part of the project, where previous solutions that are the same on all boards in KTHFS will be reused on this hardware as well.

II. EVALUATION OF APPROACHES

The problem stated above can be solved in various ways. The biggest difference between different solutions is how the user interacts with the system, i.e. what HMI to use. One possibility is to use a numerical pad to input data and then use LEDs with labels that output information to the user. This would be easy and efficient, although it would severely limit further development and improvement of the system since hardware would need to be added in order to implement new features.

Another possibility is using a touchscreen. This would mean that the user could both input data and read data from the screen. Using a touchscreen provides an easy interface and the possibility to improve the system by only writing additional software, which is typically easier compared to adding hardware. The touchscreen approach was chosen for this project because of this reason.

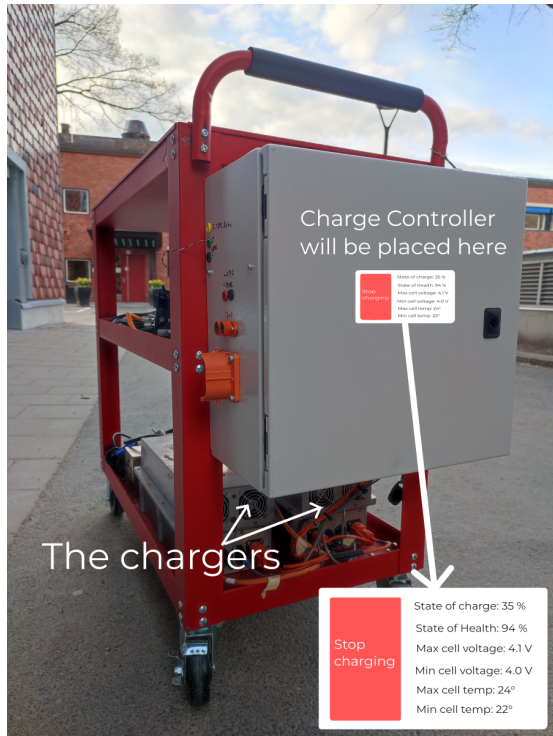


Fig. 1. The wagon housing the chargers showing where the Charge Controller will be placed.

III. LIST OF ACRONYMS

Below is a list of the most used acronyms in this thesis report.

AMS	Accumulator Management System
CAN	Controller area network
CC	Charge controller
DBC	CAN database file
DRC	Design rules check
dev-board	Development board
ECAD	Electrical computer-aided design
ERC	Electrical rules check
GUI	Graphical user interface
HMI	Human machine interface
IC	Integrated circuit
KTHFS	KTH Formula Student
LSB	Least significant bit
MCU	Microcontroller unit
MSB	Most significant bit
PCB	Printed circuit board

IV. THEORY

What follows is a description of the various parts of the project. Starting with hardware related theory and then transitioning over to the software side of the project.

A. PCB design process

In order to design a PCB an Electrical Computer-Aided Design (ECAD) program is needed. There are several programs available, both open-source and proprietary ones. The steps

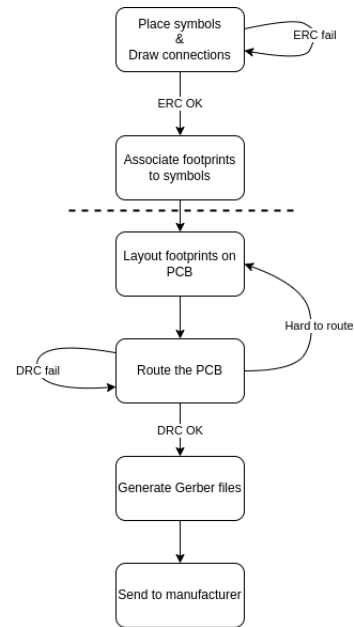


Fig. 2. PCB Design process block diagram.

in designing a PCB using an ECAD program is described in detail below and are also outlined in Fig. 2. The reader should note that this process may vary from project to project and is somewhat subjective. The steps above the dotted line in Fig. 2 are usually referred to as "schematic entry" and they take place in a different program than the ones below the line which handle the layout and routing of the PCB.

1) *Placing symbols and drawing connections*: First, one needs to design the schematic of the circuit. This step consists of placing symbols, which can be any arbitrary component such as a connector, a resistor, an IC chip, etc. This step also consists of connecting all the pins of all components to where they should be connected.

When the step above is complete, it is good practice to run an Electrical Rules Check (ERC) on the schematic. This checks whether there are for example pins left unconnected or if the designer accidentally made a mistake and maybe shorted power to ground. This can function as a major time saver since ordering a faulty PCB consumes a lot of time and resources.

2) *Layout footprints*: When the circuit is considered done and the ERC shows zero errors, the next step is to actually layout the components of the PCB. In order to do this, the designer first needs to make sure that every symbol has an associated footprint. The footprint is the pad that will end up on the produced PCB and where the component will be soldered, as seen in Fig. 3 where the footprint of a capacitor is shown. The choice of footprint is crucial since choosing the wrong footprint often means it will not even be possible to solder the component onto the board afterwards.

When laying out footprints it is typically beneficial to place components that are close to each other in the schematic close to each other also in the layout. One also needs to take into consideration that there should be as little overlap between the wire connections as possible, since this will significantly ease the process of routing the PCB which is described below.



Fig. 3. Example of a footprint.

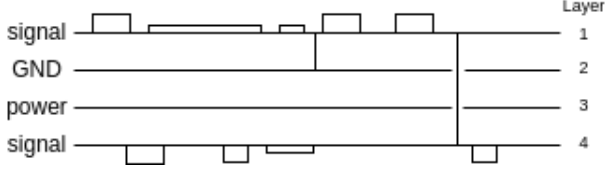


Fig. 4. Cross section of a typical 4-layer PCB.

3) *Route the PCB*: When components have been placed onto the board, the designer needs to draw all the traces on the different layers, an example of the layers on a four-layered PCB can be seen in Fig. 4. This step of the process is often referred to as "routing", which refers to laying out tracks on the board. As assistance, the program highlights where the connections needs to be done so the designer does not need to remember what connects to where. Note that tracks and traces are two words that are often used interchangeably since they refer to same thing.

When the circuit is routed, it is good practice to run a Design Rules Check (DRC), which checks that the routing follows the limits set by the user in the program [3].

However, the reader should note that routing of PCBs are a highly intricate topic, with many nuances. Generally though, what the PCB should do ultimately decides what aspects that need to be considered. A high frequency circuit will need a lot of considerations whilst a simple circuit with no high speed or power-heavy components does not need as much consideration.

When routing, it is often the case that the designer needs to go to a different layer in the PCB. This is accomplished by using vias, which can be seen as the vertical lines that connect one layer to another in Fig. 4.

Traces that carry a significant amount of current need to be sufficiently wide so that they do not overheat and melt. And it is also important to take into consideration the capabilities of the PCB manufacturer. For example, manufacturers are usually only able to make traces within specific margins [4].

B. Bypass capacitors

Bypass capacitors are used in practically every circuit, and are important in guaranteeing a clean DC power supply voltage. When AC enters the circuit from some unknown noise source, the bypass capacitor begin acting as a short-circuit as can be seen in the following equation which describes the reactance of a capacitor:

$$X_c = \frac{1}{2\pi fC} \quad (1)$$

As the frequency f increases, the reactance X_c decreases and vice versa. This means that high frequency AC noise

induced in the circuit will get shorted to ground instead of for example entering a microcontroller.

Placement of bypass capacitors is also of great importance when designing a PCB. Ideally, they should be placed as close to the relevant pin as possible. If the capacitor is far away, noise can seep into the remaining length of the trace which is undesirable.

Bypass capacitors also function as a local power supply to the nearby IC, as many ICs are driven by a clock and when the switching of transistors inside the IC occur, large amount of current are drawn in a very narrow time window. This is another reason to place the bypass capacitors nearby the ICs pins.

Often, two capacitors are used, one with less capacitance and one with more. The one with more capacitance can store more energy at the compromise of slower delivery of the energy, whilst the one with less capacitance can deliver small amounts of energy but quickly [5].

C. The chargers

Two identical chargers connected in series are used in this project. They are made by the company Current Ways with model number "CA11H03-8010". Each individual charger is capable of delivering 225 V - 450 V, 3 kW. They have been setup to work in series and they handle configuration of who should be master and who should be slave at startup. This means that two chargers in series are able to output 450 V - 900 V, 6 kW.

The manufacturer has provided an application note, which can be found at Appendix A, that describes how the entire CAN structure of the chargers work as well as a user manual, that can be found at Appendix B, that explains how to use the chargers. CAN is described in detail in Section IV-D below. In Appendix A, there are a few messages that are essential for using the chargers, these are listed in Table I which are described in detail in the following paragraph.

Byte 0 can be seen as a multiplexer-byte that the charger utilizes to decide how to interpret Byte 1, 2 and 3. The first row in Table I is the voltage request message and MSB and LSB refers to one single hexadecimal number. For example, if MSB = 01 and LSB = 02, the number is 0102 in hexadecimal, which is 258 in decimal, and since there are two chargers the total output voltage becomes the double, i.e. 516 V. The "Set current" message is for maximum current output and "Set power" is for maximum power output. The "Power on" is used to either enable the charger for YZ seconds or disable the charger, it is the X bit that determines whether it is an enable or disable message. In order to actually turn on the chargers using the "Power on" message, it is required that all of the above messages have been sent before, i.e. "Set voltage", "Set current" and "Set power". Sending for example X = 1, Y = 0 and Z = 1 means the chargers will turn on for 1 second before automatically turning themselves off.

D. CAN - Controller Area Network

CAN is a communication protocol for communication between MCUs, developed by the company Bosch. It is widely

TABLE I
CAN FORMAT FOR CHARGER MESSAGES, VALUES ARE IN HEX.

Description	Byte 0	Byte 1	Byte 2	Byte 3
Set voltage	40	FF	MSB	LSB
Set current	41	FF	MSB	LSB
Set power	42	FF	MSB	LSB
Power on	44	FF	0X	YZ

used in the automotive industry due to its simplicity and robustness, lately it has also been used in other industries as well. Several systems are connected to what is called a CAN-bus. Messages are then broadcasted on the CAN-bus and can be read by all nodes connected to the bus. In other words, a message cannot be addressed to a specific node, instead, all nodes connected to the bus receives the message and decides on a hardware or software level if the message is of relevance for the node [6].

1) *Physical layer*: The physical layer of the system consists of two wires, CAN-high and CAN-low, connected in parallel and terminated with a 120Ω resistor at both ends, as seen in Fig. 5. CAN-high and CAN-low together make a differential pair referred to as the CAN-bus and is where the CAN messages are transmitted. The voltage of CAN-high and CAN-low is approximately 2.5 V when a 0 bit is transmitted. Transmitting a 1 on the bus means that CAN-high rises to 3.5 V while CAN-low is decreased to 1.5 V. The resistors at each end are called termination resistors and the resistance across CAN-high and CAN-low shall be approximately 60Ω in order to keep the CAN-bus stable. Every node on the CAN-bus is also connected in parallel with the bus as in Fig. 5.

In order for the MCU to be able to decode the bits on the CAN-bus a CAN-transceiver IC-chip and a CAN-controller has to be connected between the bus and the MCU. Some MCUs has a built in CAN-controller, such as the STM32F7.

The speed of the CAN-bus can be up to 1 Mbit/s and is dependent on the length of the bus and the capabilities of the devices connected to the bus [7]. This is also known as the baud rate in communication which has the unit symbols / second. For digital systems, where the signals are binary the baud rate is the same as bit per second. The baud rate of the CAN peripheral for the MCU is dependent on the MCU clock speed and some timer settings for the CAN peripheral. In order to have the correct baud rate, one has to calculate the specific timer settings after the clock speed of the MCU has been set. There are calculators on the web for this purpose, such as [8].

2) *The CAN messages*: There are four different types of messages that can be sent on the CAN-bus, they are data frame, remote frame, error frame and overload frame [6]. In this thesis report, only data frame and error frame will be covered. A CAN frame consists of several fields for various information.

For this thesis report the interesting fields of a data frame message are the arbitration field and the data field which can be seen in Fig. 6. The arbitration field is the identifier of the node. The identifier of the currently transmitted message is read by other nodes on the bus. That way the receiving nodes can

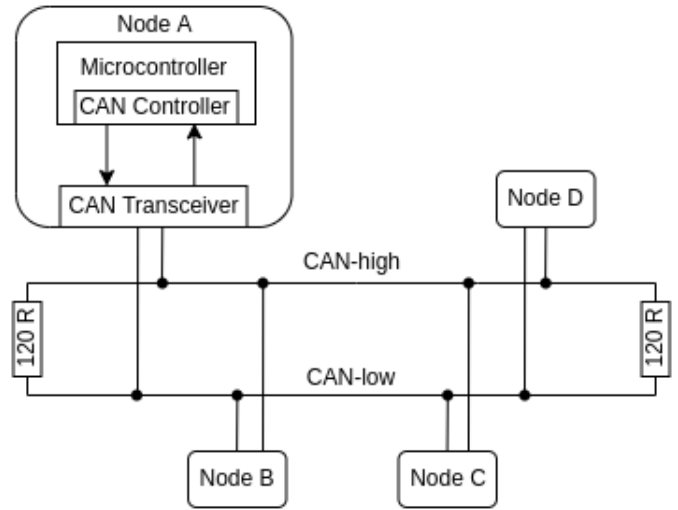


Fig. 5. CAN physical model.



Fig. 6. CAN data frame [6].

determine if the message is of importance for the respective node. It is possible to abuse the identifier, for example "node A" could transmit a message using the identifier of "node B" and every node on the bus would then believe that "node B" actually sent the message. For CAN 2.0A the identifier is 11 bits [6].

In a data frame message the data field is where the actual data or signal resides. It can be between zero to eight bytes long, i.e. up to 64 bits [6]. The data field can contain several signals, for example, voltage, temperature, speed, etc. The signals are typically decoded with the help of CAN database files (DBC), which will be covered further down.

As soon as a node on the CAN-bus notices that a message is corrupted, the CAN-controller of that node is designed to transmit an error frame message on the bus. The error frame will then trigger the other nodes on the CAN-bus to also send error frames. This will cause the initial node that sent the corrupted message to automatically try to re-transmit the message. If the message is again corrupted the CAN-bus will once again be filled with error frames. If the node is repeatedly sending corrupted messages, the CAN-controller of that node will automatically put the node in offline mode so that the CAN-bus is not occupied with faulty messages [6].

3) *DBC files*: In order to decode the raw CAN data on the CAN-bus, DBC files are used as mentioned above. DBC files are text files containing decoding rules for the different messages. In the DBC files the different identifiers are specified with all the signals for the data field. It is specified how the raw CAN data should be decoded to physical values such as voltage, current, speed, etc. [9]. These DBC files can then be converted to C code which can be used for communication

software.

4) *Monitoring a CAN-bus*: When developing a system which communicates with CAN, it is often desirable to be able to monitor the CAN-bus in real-time. The company Kvaser provides interfaces for monitoring and sending messages on a CAN-bus. For example their product Kvaser USBcan [10]. Kvaser USBcan is connected with USB 2.0 to a computer and the CAN-bus can then be monitored on the computer with a program. One program that is capable of this is CANLab from Accurate Technologies [11]. The program, together with Kvaser USBcan is able to monitor a CAN-bus, send messages on a CAN-bus and import DBC files in order to also decode and send predefined messages.

E. GUI - Graphical User Interface

A graphical user interface (GUI) is a way for a user to interact graphically with a system through a screen. They often includes icons, text labels, software buttons, etc. The opposite of this is a text-based user interface where the interaction between human and computer is done with text commands, for example the terminal in computers.

In order to create a GUI there are several possible GUI libraries that can be used. Some popular libraries are Qt, uGFX, TouchGFX and LVGL. These libraries differ in several ways. They can for example differ in how easy they are to program. Some libraries have for example GUI editors where the GUI is designed using another graphical interface, whilst other libraries has to be manually programmed line by line. They also differ in speed and memory footprint. When designing a GUI for an embedded system it is important that the memory footprint of the GUI is small since the memory in embedded systems are often highly restricted. In this thesis LVGL (Light and Versatile Graphics Library) was used and will be explained below.

LVGL is an open source object oriented GUI library available in C and MicroPython. It is popular since it is open source, has a small memory footprint and can run on several platforms such as STM32, Arduino, Raspberry Pi, etc. [12].

Everything displayed on the screen is based on LVGL objects. Objects are placed on other objects and can be seen in Fig. 7 how a simple screen could look like. To create a screen that can yield buttons and text labels one has to create a new `lv_obj_t *` pointer with the `lv_obj_create()` function and `NULL` as argument (essentially a parent) to `lv_obj_create()`. In order to create a button on the newly created screen, a new `lv_obj_t *` has to be created but this time with the `lv_btn_create()` function and now the screen as parent to the object. That way the button will be housed on the screen. The button can now have a label with the button as parent. In this way a GUI can be built.

In order to be able to interact with the GUI, callback functions has to be implemented. Callback functions are executed when a trigger event occurs. In LVGL and other GUI libraries a trigger event can for example be a press or a click on the screen. These callback functions can on the other hand trigger other code blocks in several ways, such as toggling an LED, set a flag variable to notify the program that a certain condition

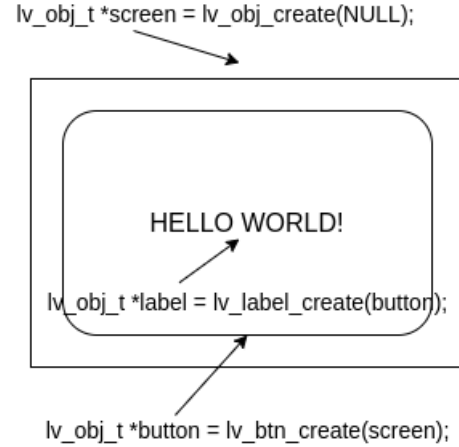


Fig. 7. Diagram showing how LVGL objects are placed on top of each other.

has been met or load a new screen. To summarize, in order to have a button react to a press on the screen, a callback function has to be added to the specific button object. When the button on the screen is pressed this will cause the callback function to execute.

F. STM32CubeIDE

The software has been written using STM32CubeIDE which is an integrated development environment for the STM32 family of MCUs. It is based on the widely used development environment Eclipse. In STM32CubeIDE it is possible to graphically configure the input and output peripherals of the MCU. STM32CubeIDE can then auto generate code for the required configuration [13].

G. AMS - Accumulator Management System

The Accumulator Management System (AMS) is a system on the KTHFS vehicle closely related to the battery. A more widely used term is Battery Management System or BMS, but within KTHFS the system is called AMS. The battery consists of 6 segments, where each segment consists of 21 cells connected in series and each "cell" is actually two cells parallel connected. The cells are lithium-polymer. The purpose of the AMS is to monitor, among other things, the voltage and temperature of all the cells in the battery.

V. METHOD

A. System architecture

In the early stages of the project, several system designs were considered. It was important to lay a solid foundation of how the CC was going to interact with the other systems and what was necessary on a hardware and software level. The different systems involved can be seen in Fig. 8. The system got a verification from KTHFS members.

The charging system architecture includes three systems of the KTHFS vehicle. The chargers, the AMS and the CC. They are all connected to a CAN-bus with the speed 500 kbit/s. The idea was that when the user would press "Start charging" on the CC, the CC would send a CAN message to the AMS to

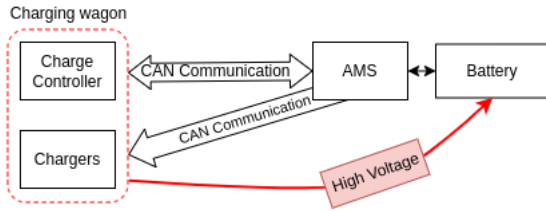


Fig. 8. Overview of the systems involved.

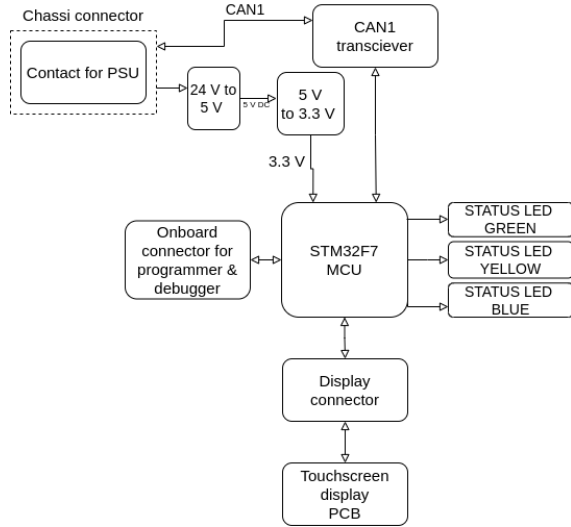


Fig. 9. PCB Block diagram.

notify that the AMS shall proceed with charging. The AMS would then perform some diagnostics on the battery before proceeding with charging. When the AMS was ready it would then send CAN messages to the chargers and ask for the desired voltage and current.

B. Hardware

1) *Block diagram*: In order to design the hardware platform, a block diagram was first setup with only the essential components of the circuit. This block diagram can be seen in Fig. 9. In order for the CC to perform its function successfully, these were the required parts. The main focus was on implementing these blocks in the start, and if there was time left over, more functions would be incorporated into the circuit. For example a circuit for measuring temperature with the help of thermistors, as well as physical buttons and LED indicators on the enclosure of the CC.

As can be seen in the middle of Fig. 9 the chosen MCU for the project was an STM32F769IIT6. It is a powerful MCU with 176 pins, and a large enough memory for the screen buffer. It is the same MCU that was on the development board (dev-board) which can be seen in Fig. 10.

2) *Design process*: The ECAD program KiCAD was chosen for this project. The main reason being that KTHFS already uses it for all the other circuits done for the car. As many PCBs already has been done before, many circuits were reused in the CC schematic in order to save time and increase reliability since those circuits were well tested.



Fig. 10. STM32F7 development board used for testing.

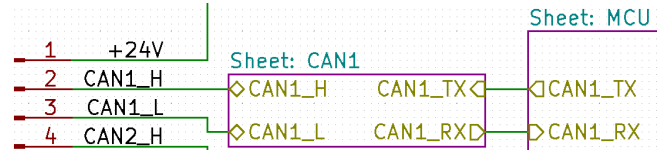


Fig. 11. KiCAD CAN1 Sheet.

The PCB block diagram shown in Fig. 9 was used to separate the circuit into several KiCAD sheets, where one sheet ideally contains components only related to some specific function of the circuit. For example, the DC-DC converters are sheets that abstract away the implementation of the converter. The user of the sheet are then only presented with one input and one output from the sheet, in this case 24 V as input and 5 V as output. The usage of sheets simplified the design and made it easier to overview the different parts of the circuit.

An example of a sheet from the finished design is shown in Fig. 11. The connector comes from the left, enters the CAN sheet and performs a translation of the differential CAN signal to a single-ended signal that the MCU can interpret. Notice that the MCU also has its own sheet. The benefit is clear here, the person designing this schematic does not need to know the exact details of how the translation from differential to single-ended signal works, it is sufficient to know that it will take care of it and what the sheet expects and what it outputs. The translation might be implemented in various ways, either using an application specific integrated circuit (ASIC) or maybe with discrete components, and it is this fact that efficiently enables a different person to be working on the CAN sheet and not have to worry about the rest of the circuit.

3) *PCB Stackup*: A 4 layer board was chosen for this project with the layers arranged as seen in Fig 4, mainly because the manufacturer JLCPCB offered cheap 4 layer boards and that routing becomes much easier on a 4 layer board since it is always possible to access both GND and power simply through a via connection. However, another crucial point is that 4 layer PCBs are better at both not picking up electromagnetic noise as well as generating electromagnetic noise.

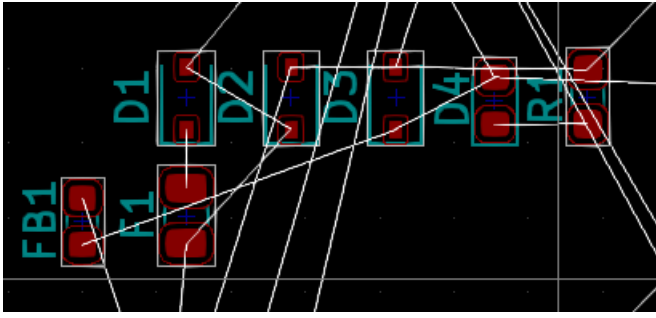


Fig. 12. KiCAD Air wires.

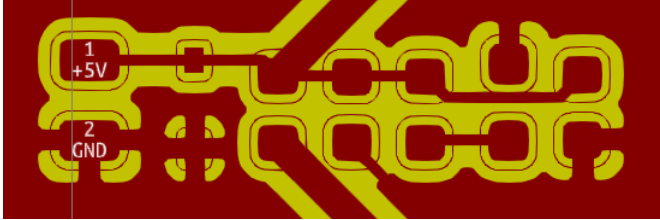


Fig. 13. Example of routing, red color means copper trace, yellow color means non-conductive.

4) *Routing*: When the above step was done, routing began. Routing was done by using KiCADs built in tool for routing PCBs. Routing consisted of drawing traces between all the components. Routing this board proved difficult and took several tries before the routing was complete. One of the challenges was placing the components in a manner that minimized different wires overlapping with other wires.

The first step of routing was to import the schematic into the PCB layout tool, and Fig. 12 show a small example of how it looked just after importing. The white lines are lines that tell where the connection should be made to and it is important to then line up the components in such a way that there is minimum overlap between the white so called "air-wires". Spending lots of time rotating and aligning components in a logical manner proved to be very efficient at minimizing the number of vias on the board and making the routing easier.

Once the placement of all the components was finished, the traces were routed. Special attention was given towards, for example, the power supply traces that will carry much more current than merely signal traces. Power supply traces were made wider. See Fig. 13 to see the components in Fig. 12 realigned and routed. The wide traces in the middle are power traces. The color red means that it is copper and because a copper pour has been made, copper is everywhere where there are no traces and is connected to the ground layer via vias.

C. Bypass capacitors

Bypass capacitors was placed onto the circuit as per the STM32F7 datasheet which can be found in Appendix C, which explicitly says how many it should be and what capacitance they should have. They were placed as close as possible to the pins of the MCU.

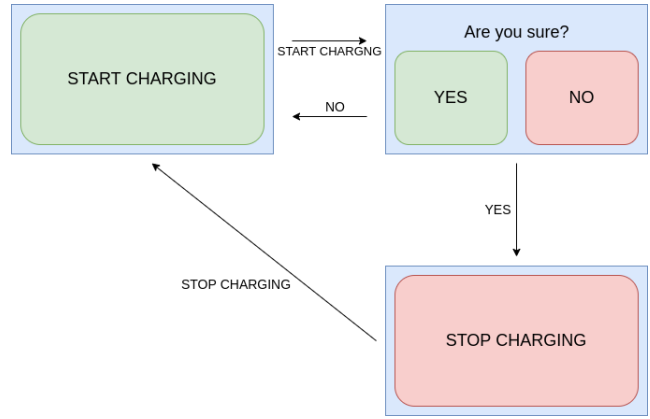


Fig. 14. First diagram of displays.

D. GUI implementation

The first step taken in the designing of the GUI was to sketch a first draft for a block diagram of the different screens and the events that causes the transitions between them. The sketch can be seen in Fig. 14.

Early on, the decision was made to write the GUI using the GUI library LVGL. This decision was primarily based on the fact that LVGL had previously been successfully used and ported to the STM32F7 MCU within the KTHFS team, which was the same MCU used during this project. Another factor was because the memory footprint of LVGL is small.

In the beginning the focus was to understand and experiment with LVGL. Buttons and labels were created and uploaded onto the dev-board, which was the board that was used to test the initial code for the project. These were then after trial and error, successfully displayed on the screen and further development could proceed. One milestone that was achieved was to make a button press on the screen make an LED turn on. This indicated that a button press on screen was successfully registered and was capable to interact with other parts of the hardware. This LED toggle could then in theory be replaced by for example with sending a CAN message.

When basic knowledge of LVGL was achieved, the screen was implemented according to the diagram in Fig. 14. After this, it was a matter of merging the screen software with the communication software which is explained later in Section V-F.

E. CAN communication

1) *Configure CAN*: In order to enable and use the CAN peripheral the code had to be set up to support it. The process had been done several times before within the team. This meant that a lot of the code was already created and needed to be pasted into the STM32CubeIDE project. What needed to be done for this project was to enable the CAN peripheral on the MCU and also calculate the timer settings in order to send the CAN messages on the right baud rate, i.e. 500 kbit/s. These calculations were done using [8], which is an online tool for calculating the clock parameters for the desired baud rate.

When the STM project was configured to be able to send CAN messages, the dev-board was connected to a CAN

transceiver. In order to test whether CAN communication was successful, Kvaser USBcan together with CANLab software was used to monitor the traffic on the CAN-bus.

2) *Sending CAN messages:* Sending CAN messages on a CAN-bus was first done with a STM32F0 Nucleo board which is a simpler board than the STM32F7 dev-board. This was done in the beginning when the goal was to get familiar with how CAN works. When using the STM32F0 Nucleo, board the messages were correctly transmitted on the CAN-bus and displayed on the CANLab software.

Problems occurred when sending CAN messages from the STM32F7 dev-board, which has the same MCU that the designed PCB uses. In order for the screen to not flicker, the system clock speed had to be high, around 200 MHz. Even though the clock calculations for the 500 kbit/s CAN baud rate were correct, the CAN-bus was flooded with CAN error frames. The messages sent on the CAN-bus seemed to be corrupted. Further experimentation with the system clock speed made it clear that when the system clock speed was above 140 MHz, the messages being sent on the bus were being corrupted. On the other hand if the system clock speed was below 200 MHz the screen was not displayed properly. After discussion with the supervisors it became clear that this was due to the fact that the system clock was controlled from the high speed internal clock. The problem with the high speed internal clock seemed to be that when the frequency was high, the clock started to become imprecise. The solution was to instead use the external crystal oscillator which was more exact. When the external crystal oscillator was used, the CAN messages were transmitted and correctly read on CANLab.

3) *Receiving CAN messages:* When sending CAN messages was possible, it was time to configure the system to be able to receive messages. This was done by configuring interrupts for whenever the MCU received a message. Whenever the system received a message, the program would get interrupted and unpack the message. The message would then be read and depending on what the message was it would behave differently. In order to test if the MCU properly received the CAN message, a test was conducted where the right message would trigger the MCU to turn on an LED. This test was successful.

4) *Extending the DBC files:* Now that the system was able to send and receive CAN messages it was time to implement a few messages in the DBC files of KTHFS. This was done through a software called Kvaser Database Editor 3. To the existing DBC files of KTHFS, the messages for the chargers were defined according to the manufacturer specification which is described in Section IV-C and in Table I. Since the chargers were connected in series the power and the voltage had to be prescaled by 0.5 in the DBC files.

A few other messages were defined for the system. A message called `cc_status` was created and the purpose of this message was to be able to notify the AMS if it shall proceed with the charging. There was also a new signal added to the existing message called `ams_status_1`. The purpose of the new signal to the message was to be able to indicate the charging status of the AMS, for example if it was charging, discharging or being idle.

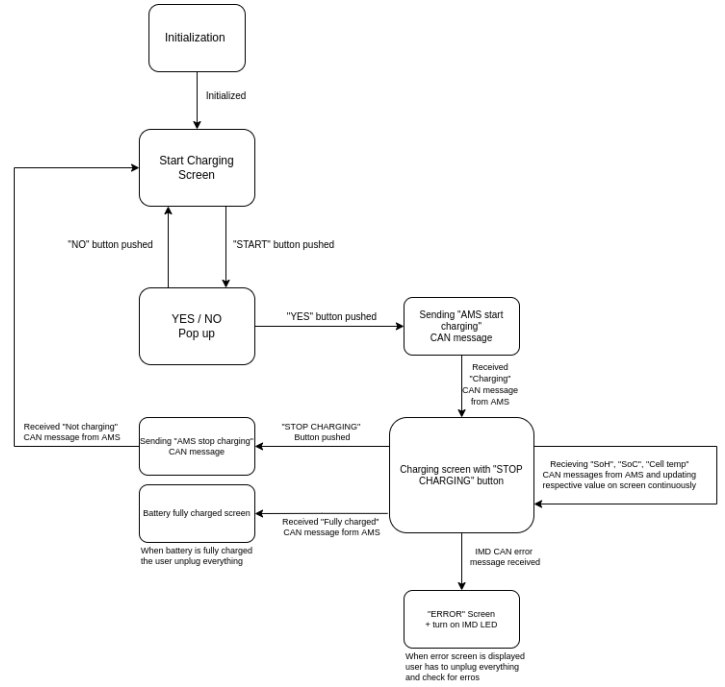


Fig. 15. State diagram of software.

When all the messages had been defined in the DBC files, they were committed and pushed to the KTHFS GitHub repository. The repository then generated C code from the DBC files using its own automatic tool within GitHub. The C code were then included into the STM32 project in order to be able to use the new messages.

5) *Communication software:* The next step was to write code where the newly defined CAN messages were utilized. The messages were sent on a CAN-bus and monitored with the Kvaser USBcan tool and CANLab. Initially some of the DBC files were not defined correctly so minor changes had to be made but after trial and error they looked as desired. The last step was to develop an easy to use library for sending CAN charger messages.

F. Merging GUI with CAN communication

When the GUI, DBC files and the CAN structure was done, it was time to integrate the GUI with the communication part of the software. In order to do so, a state diagram for the software was designed as can be seen in Fig. 15.

The structure of the GUI had to be updated substantially to fit the state diagram. Initially the GUI was setup in a way that made adding functionality quite challenging. The code was nested in complicated ways and therefore had to be redone.

A function called `screen_handler(int CHOICE)` was defined instead and the purpose of this function was to load all the different screens depending on an argument passed to `screen_handler`. Every screen available in the GUI could be loaded from the `screen_handler`. This function would be called depending on what CAN message was received according to the diagram in Fig. 15. A few more screens were created for indication when errors occurred and

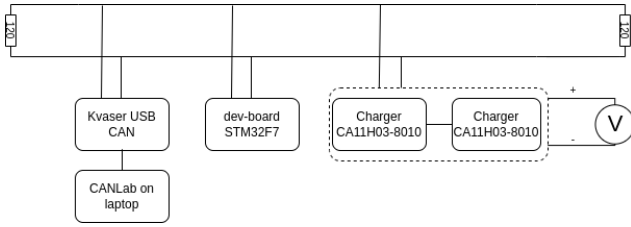


Fig. 16. Showing the systems involved in the test setup .

Msg Id [hex]	Msg Name	Data [hex]
0x250	cc_status	01
0x020	ams_status_1	01 00 00 00...
0x270	charger_config	40 FF 00 E1
0x270	charger_config	41 FF 00 0A
0x270	charger_config	42 FF 00 64
0x270	charger_config	44 FF 01 14
0x270	charger_config	40 FF 00 FA
0x270	charger_config	44 FF 01 14
0x270	charger_config	40 FF 01 13
0x270	charger_config	44 FF 01 14
0x270	charger_config	44 FF 00 00

Fig. 17. CAN-bus from testing the chargers.

when the battery would be fully charged. When the structure was re-written it was tested with Kvaser USBcan to verify that the software was behaving according to the state diagram.

VI. RESULTS

A. Live testing with the chargers

A test has been performed using the dev-board, which successfully controls the chargers in the desired manner. Fig. 16 shows the test setup in a diagram. The test setup consists of the following equipment:

- Charging station wagon
- 2x CA11H03-8010 charger connected in series
- Multimeter set to show DC voltage
- Computer running CANLab 5.0
- Kvaser USBcan
- dev-board

CANLab allows us to both view the data sent on the CAN-bus in real-time and transmit messages. This is very convenient because it means the AMS does not need to be used in the test setup, instead, it is possible to emulate it using CANLab, sending the messages the CC is expecting from the AMS. The charging button "Start charging" was pressed on the CC and on CANLab the CAN messages could be read, as seen in Fig. 17.

In order to test that the chargers output the desired voltages, a code sequence was written that asks for 450 V, 500 V, 550

TABLE II
TEST RESULTS FROM TWO TESTS

Time [s]	Requested voltage [V]	Measured DC voltage [V]
0	450	451.7
10	500	501.6
20	550	551.7
0	450	451.8
10	500	501.8
20	550	551.9

V with 10 seconds between each change. The results of this test can be seen in Table II.

Fig. 17 shows the actual data sent on the CAN-bus, the first message with ID = 0x250 is the message that the CC sent in response to a human pressing the "Start charging" button which tells the AMS to start the charging procedure. When the AMS (in this case Kvaser USBcan) has performed its own diagnostics and begin charging, it transmitted a message with ID = 0x020, telling the CC that charging will begin. That resulted in the CC screen being updated to show the screen where the user can press "Stop charging" button. The charging commands were then sent to the chargers, which are all the ID = 0x270 messages. More details regarding the exact meaning of the messages with ID = 0x270 can be found in Section IV-C of this report. But in essence, they simply configure the voltage, current and power of the chargers and tells them how long to remain on.

B. Communication

The software together with the dev-board and a CAN transceiver is able to communicate with systems on a CAN-bus. The DBC files has successfully been extended to support new features and communication with the chargers, the CC and the AMS. They have been tested and are working as expected, as mentioned in Section VI-A. Although they have not been tested with the actual AMS but with the Kvaser USBcan as a substitute.

CAN communication has not been tested and verified for the designed PCB.

C. Screen software

A simple to use GUI has been successfully developed for the dev-board. Pictures of the different screens can be seen in Fig. 18,19 and 20. The "Start charging" button was moved to the upper half of the screen in order to minimize the risk of accidentally start charging, the user has to move the finger in order to confirm the choice between Fig. 18 and 19. The i Screen for notifying error and fully charged battery has also been implemented but pictures has not been included in the report. It is able to change screen depending on what CAN message has been received on the dev-board.

The screen software has not been tested and verified on the designed PCB.

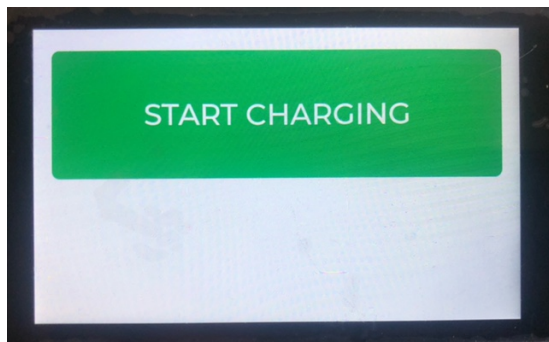


Fig. 18. Photograph of screen displaying "Start charging" button.

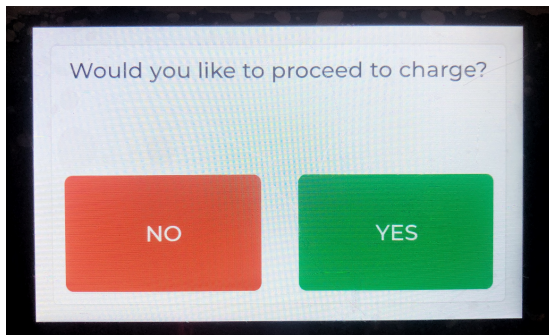


Fig. 19. Photograph of screen displaying choice to proceed with charging.

D. Hardware and PCB

The PCB was successfully designed and has passed both ERC and DRC tests. A render of the PCB has been done and can be seen in Fig. 21 and how the PCB looks like in KiCAD can be seen in Fig. 22. It was sent for manufacturing at the company JLCPCB. The retrieved PCB can be seen in Fig. 23. Continuity tests was performed on the PCB to make sure that there were no obvious problems in the produced PCB such as short circuits between power and ground.

Soldering of the PCB is partially done and the current state of the PCB can be seen in Fig. 24. The MCU has booted successfully and code has been uploaded to it. The code toggled three of the LEDs on the PCB which shows that the MCU works.

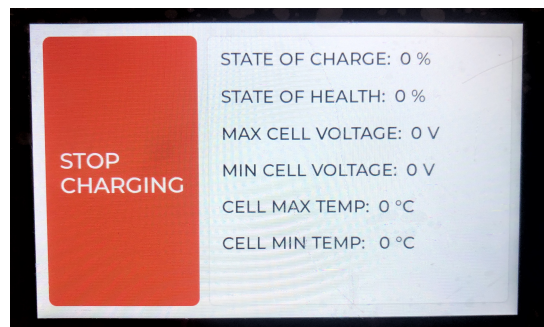


Fig. 20. Photograph of screen displaying "Stop charging" button.

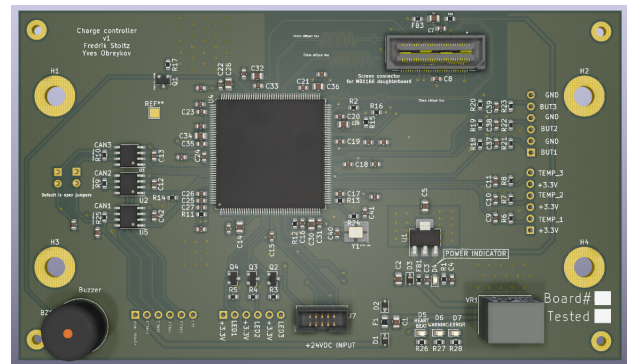


Fig. 21. Render of PCB.

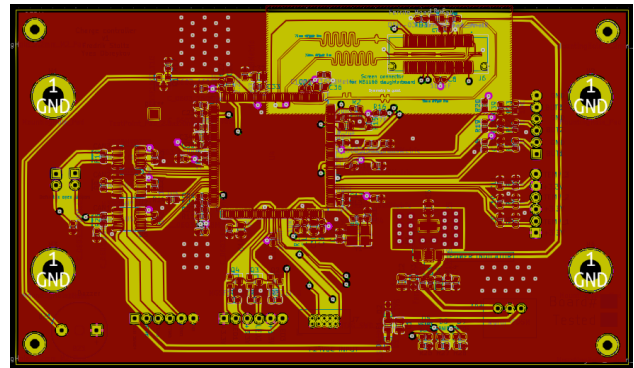


Fig. 22. The PCB in KiCAD.

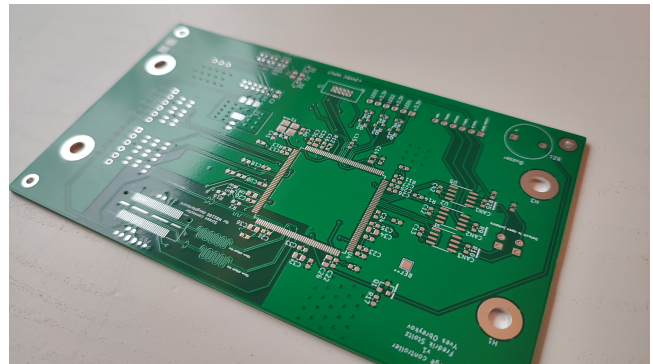


Fig. 23. Manufactured PCB.

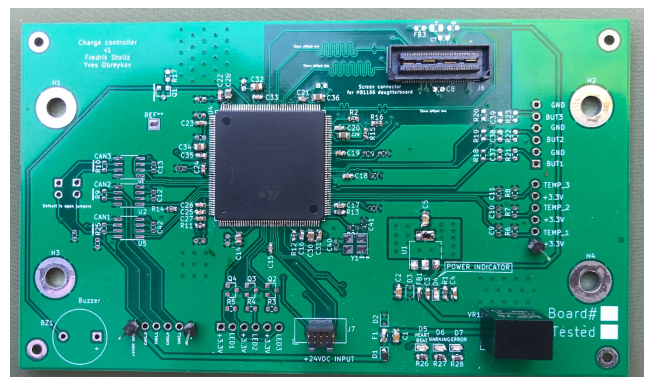


Fig. 24. Partially soldered PCB.

VII. DISCUSSION AND CONCLUSION

The result from the live testing mentioned in Section VI-A proves that communication with the chargers are reliable and work as intended. The prototype has not yet been tested together with the actual AMS, where the charging commands are supposed to reside, but by nature of the CAN protocol it should in theory not depend on from what system the charging commands are sent from. As can be seen in Fig. 18, 19 and 20, the GUI is simple and easy to understand. We believe that the foundation laid within this thesis project works as an excellent starting point for further development of the system.

The designed PCB has been partially soldered and the MCU has been tested, although not with the touchscreen. Since the system is largely based on another system in KTHFS that works without any problems, we are confident it should not be any major issue integrating the touchscreen. Further more, continuity tests shows that there should not be any faults in the routing of the PCB and it has passed every software error check. This indicates that it should work properly when all of the components have been soldered.

The choice of MCU was an STM32F769IIT6, which in hindsight was an unnecessarily powerful MCU for this task. More than half of the peripherals are not in use. We could have spent more time on evaluating alternatives before committing to a specific MCU, although there was only one feasible MCU for our purpose in stock at KTHFS and therefore we choose the STM32F769IIT6. An alternative solution that should have been considered would have been to use an STM32F4 together with an external memory for the screen. This would have been an easier PCB to solder and a resource efficient solution, but because of the current shortage of semiconductors it is almost impossible to order new STM32F4 MCUs.

To conclude this thesis report, the project has laid the foundation of a reliable HMI for simplifying the charging process of the KTHFS vehicle. It is flexible and has plenty of room for future improvements and the authors of this report plan on continuing the work on the Charge Controller.

VIII. FUTURE WORK

There is still a lot of improvement possibilities that are listed below. The hardware for some of these possibilities are already implemented in the AMS, so for certain points it is only a matter of extending the capability on the software level but also some hardware-related work in regards to the physical CC.

- Testing the hardware
- Designing and printing an enclosure for the CC
- Integrating the enclosure into the gray box as seen in Fig. 1.
- Developing software that will display charging metrics on the CC screen
- Placing sensors in the gray box seen in Fig. 1 that measure temperature
- (Thesis) Implement the software as a Real-time operating system
- (Thesis) Implement various charging modes such as fast charging and slow charging.

We believe the points marked as (Thesis) are suitable for a future Bachelor thesis project. Note that implementing the software as a Real-time operating system is only reasonable if the software would gather data from the AMS and have more requirements on it than it currently has.

APPENDIX A

CAN STRUCTURE OF THE CHARGER

APPENDIX B

USER MANUAL OF THE CHARGER

APPENDIX C

DATASHEET FOR STM32F7 MICROCONTROLLER

ACKNOWLEDGMENT

The authors would like to thank supervisor Carl-Mikael Zetterling for his help throughout the project and the whole of KTHFS for assistance and discussions. Matthias Becker and Mark Smith have also contributed to the project which we are grateful for.

REFERENCES

- [1] KTH Formula Student. (2022, Apr.) Kth formula student. [Online]. Available: <https://kthformulastudent.se/>
- [2] H. Othman, Y. Aji, F. Fakhreddin, and A. Al-Ali, "Controller area networks: Evolution and applications," in *2006 2nd International Conference on Information Communication Technologies*, vol. 2, 2006, pp. 3088–3093.
- [3] A. A. Gautam and V. Laxmi, "Gate drive for power electronic converters : An insight into kicad's pcb design !," in *2021 2nd Global Conference for Advancement in Technology (GCAT)*, 2021, pp. 1–6.
- [4] Autodesk. (2022, Apr.) Everything you need to know about trace width. [Online]. Available: <https://www.autodesk.com/products/fusion-360/blog/trace-width/>
- [5] M. I. Montrose, "Bypassing and decoupling," in *EMC and the Printed Circuit Board*, ser. IEEE Press Series on Electronics Technology. New Jersey: John Wiley Sons, 2005, vol. 9, pp. 125–158.
- [6] Kvaser AB. (2022, Apr.) The can bus protocol tutorial. [Online]. Available: <https://www.kvaser.com/can-protocol-tutorial/>
- [7] M. Desai, R. Shetty, V. Padte, M. Parulekar, and S. Ramrajkar, "Controller area network for intelligent vehicular systems," in *2013 International Conference on Advances in Technology and Engineering (ICATE)*, 2013, pp. 1–6.
- [8] Heinz-Jürgen Oertel. (2022, Apr.) Can bit time calculation. [Online]. Available: <http://www.bittiming.can-wiki.info/>
- [9] CSS Electronics. (2022, Apr.) Can dbc file explained - a simple intro. [Online]. Available: <https://www.csselectronics.com/pages/can-dbc-file-database-intro>
- [10] Kvaser. (2022, Apr.) Kvaser usbcan light 2xhs. [Online]. Available: <https://www.kvaser.com/product/kvaser-usbcan-light-2xhs/>
- [11] Accurate Technologies. (2022, Apr.) Canlab software. [Online]. Available: <https://www.accuratetechnologies.com/Products/CANLabSoftware>
- [12] LVGL. (2022, Apr.) Light and versatile graphics library. [Online]. Available: <https://lvgl.io/>
- [13] STMicroelectronics. (2022, Apr.) Integrated development environment for stm32. [Online]. Available: <https://www.st.com/en/development-tools/stm32cubeide.html>

A General Purpose Analog Circuit to Accumulate Data From Resistive Sensors

Alan Alhallak and Karl Höjlund

Abstract—Minimizing the need to physically adjust hardware platforms used for sensor measurements during the construction phase of an electric vehicle can be beneficial. Since different sensors have different measuring ranges a hardware platform used for a specific sensor might not work for another one, without physically tampering with it. One way to solve such an issue is to build a general hardware platform that can be adjusted digitally through software to match the range of a variety of sensors. In this thesis, the implementation of a prototype General Purpose Data Acquisition Unit has been investigated. The design consists of a Wheatstone bridge implementation for measurements with resistive sensors, due to its capabilities of accurate detection of small changes in resistance. Digital potentiometers were implemented in the design to add dynamic capabilities for calibration and measurements with different types of resistive sensors through software. The proposed implementation has been tested on a preboard and built on a Printed Circuit Board. Further testing is required to better specify and evaluate the proposed implementation.

Sammanfattning—Att minimera behovet av att fysiskt justera hårdvaruplattformar för mättingsprocesser med sensorer vid tillverkning av en elektriskt driven bil kan vara fördelaktigt. Olika sensorer har olika mätområden och en hårdvaruplattform kan fungera väl för en sensor men inte nödvändigtvis för en annan utan att hårdvaran fysiskt behöver justeras. Ett sätt att lösa detta problem är att utveckla en generell hårdvaruplattform för insamling av data från sensorer som digitalt kan anpassas för att fungera med ett större utbud av sensorer. I denna rapport har en implementering av en generell hårdvaruplattform för datainsamling undersökts. Implementationen består av en konfiguration av en Wheatstone brygga för resistiva sensorer, på grund av dess förmåga att noggrant mäta små förändringar av resistans. Digitala potentiometrar användes i implementeringen för att ge möjligheten till att dynamiskt kunna kalibrera och mäta data från olika typer av resistiva sensorer genom mjukvara. Den förslagna implementationen har genomfört ett test på en preboard och monterats på ett kretskort. Fler tester krävs för att bättre kunna specificera och evaluera den förslagna implementationen.

Index Terms—Data acquisition, Wheatstone bridge, Analog conditioning, Digital potentiometer.

Supervisors: MARK T SMITH

TRITA number: TRITA-EECS-EX-2022:137

I. INTRODUCTION

Building an electric race vehicle requires a variety of sensors. There are multiple complex systems inside the vehicle such as batteries, suspension, rear and front wings, brakes, and more. Sensors are necessary for monitoring and communication purposes between different systems in the vehicle. The sensors measure physical parameters such as pressure,

temperature, and distance and translate them into electrical signals. For example, a voltage sensor in the batteries informs the driver about the state of the batteries and how much charge is left or the wheel speed sensor informs about the speed of the vehicle. During the manufacturing process, there is a need to conduct different measurements requiring different sensors. A practical way of handling this problem is by building a single platform a general purpose Data Acquisition Unit (DAU), that can digitally be adjusted to work for these different sensors.

This thesis is a part of KTH Formula Student (KTHFS), where they are currently working on a new electrical race vehicle called DeV17, which stands for a driverless electrical vehicle. Under the construction process, there is a need to connect new sensors to the vehicle for testing purposes. One quick solution to solve such an issue is by connecting the sensors to a breadboard and read data through microcontroller boards. This is not an efficient solution because it can take a lot of space and given the space the data will be less accurate, due to breadboard connections being very noisy. Therefore it is of interest to KTHFS to explore the possibility of building a general purpose DAU, which is the objective of this thesis.

Prior to this thesis, there have been different approaches used for measurements with sensors. A common configuration used for sensor measurements is a Wheatstone Bridge (WB), due to its ability to measure small changes with small errors. In the paper [1], a current-based method using a WB is tested. In this paper, a bias voltage is applied to the bridge creating two currents through each leg. Then the difference of the currents are fed respectively to a buffer which converts them to two voltages and the difference of the voltages is the final output. This method yields high accuracy, linearity, and a good common-mode (CM) cancellation. A second example would be the approach shown in the paper [2], where a displacement transducer is measured with a WB and an Instrumentation Amplifier (IA), that can in theory measure displacements on a scale of millimeters. In this thesis, a WB configuration was also investigated, where the main focus was to integrate digital components into the configuration for digital calibration of resistive sensors.

LIST OF ACRONYMS

DAU: Data Acquisition Unit
ACC: Analog Conditioning Circuit
ADC: Analog to Digital Converter
WB: Wheatstone bridge
IA: Instrumentation Amplifier
A-LPF: Active low-pass filter

MCU: Microcontroller Unit
CAN: Controller Area Network
PCB: Printed Circuit Board
CM: Common-mode
I²C: Inter-Integrated Circuit Protocol

A. Project formulation

The goal of this project was to build a DAU that can be used for different sensors. The method chosen to achieve this is by having digitally adjustable hardware. A DAU is a bridge between the physical and digital world, with the role of transporting the output of sensors (most commonly voltages or currents) to a computer, where it can graphically be displayed [3]. The figure 1 illustrates an overview of the system. A common DAU consists of:

- 1) ACC: consists of analog components that amplify the sensor's signal and filter out noise from it.
- 2) ADC: a unit that samples the analog signal that comes after the ACC turning the analog signal into a digital, which in turn can be sent to a computer.

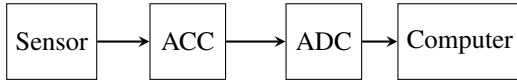


Fig. 1. Illustration of how a sensor signal is transported with a DAU to a computer, where the DAU is comprised of the ACC and ADC block.

B. Design limitations

The limitations for the design of the DAU was the following:

- Handle three to four sensors simultaneously.
- Working with a 24 Direct Current (DC) voltage supply.
- Be able to communicate via CAN.

In order to narrow down the project, the ACC was designed to only work with resistive sensors. The reason is that resistive sensors can measure many of the crucial parameters for a vehicle such as temperature, pressure, and displacement. Also taking into consideration the project's timeline and resources.

C. Approach

The difficulty with building a general purpose DAU is the ability to be configurable for a variety of sensors. Commonly, an ACC is designed to comply with specifications for the sensor that shall be used. In this project, the sensors are unspecified. Therefore the idea was to build a more general purpose ACC for resistive sensors and quantify its performance. If the ACC specification was sufficient for the area of application of the DAU, the ACC could be multiplied in order to use more than one sensor simultaneously on the DAU. The ACC is used to filter noise and match the measured range of the sensor to the input voltage range of the ADC. The analog data from the ACC is converted using an ADC-channel inside a MCU. The data measured by the MCU, can be used to calibrate the ACC. The measured value can be translated back to a physical measurement using the calibration values. The

data can be sent using CAN, which is a serial communication bus, commonly used in the automotive industry in order to minimize wiring in vehicles, since it only requires a two-wire communication bus [4]. The DAU in this project has allocated hardware space for CAN implementation but lacks the software part in the MCU.

II. BACKGROUND

The important part of a DAU is the ACC, since it is the circuit that transforms the output from the sensor to an appropriate format for the ADC. The ACC is also responsible for eliminating sensor and environmental noise. In order for a DAU to work for different types of resistive sensors, the ACC needs to have the ability to be dynamically adjusted, in such a way that the signal from the sensor connected to the ACC meets the measuring range for the ADC. The proposed solution for the DAU consists of a circuit configuration for the ACC combined with a MCU.

A. ACC

The ACC contains three parts: 1. A WB, 2. An IA and 3. A-LPF, as shown in figure 2. The mathematical model behind the ACC will be introduced in the followings order:

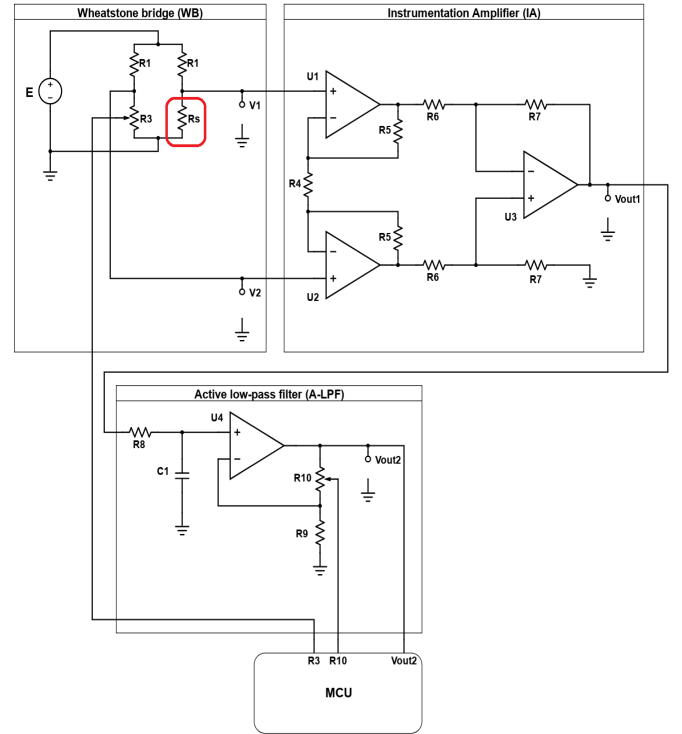


Fig. 2. A schematic of the proposed ACC where the sensor resistance R_s is marked with a red square, R_3 and R_{10} are digital potentiometers controlled by the MCU and the output V_{out2} from the ACC is sent to the MCU.

1) WB: The sensor, which is modeled by the red marked resistance R_s in figure 2, is inserted in the right leg of the bridge which is supplied with a DC-voltage E together with the two identical upper leg resistances R_1 and parallel to a variable resistor R_3 . The output of the bridge is given by

the difference of the voltage division of each leg which is mathematically written as

$$V_2 - V_1 = E \left(\frac{R_3}{R_1 + R_3} - \frac{R_s}{R_2 + R_s} \right) [\text{V}]. \quad (1)$$

The output $V_2 - V_1$ is then sent to the IA.

2) *IA*: The three Operational Amplifier (OP-amp) IA provides a good Common Mode Rejection Ratio (CMRR) and a simple way to control the gain of the input $V_2 - V_1$. The two OP-amps U_1 and U_2 provides high impedance inputs for V_2 and V_1 that buffers the input voltage $V_2 - V_1$. The input $V_2 - V_1$ will appear across R_4 . Due to symmetry, where the resistance values on the U_1 and U_2 side are equal, the gain can be controlled by R_4 alone. Also when a CM voltage is fed into U_1 and U_2 the voltage on each side of R_4 will be equal and canceled. The mathematical relation between the input $V_2 - V_1$ and the output V_{out1} from the IA, assuming that the operational amplifier's (OP-amps) U_1 , U_2 and U_3 are ideal can be derived as

$$V_{out1} = \frac{R_7}{R_6} \left(1 + 2 \frac{R_5}{R_4} \right) (V_2 - V_1) [\text{V}] \quad [5]. \quad (2)$$

The output V_{out1} is then sent to the A-LPF for filtering and amplification.

3) *A-LPF*: The active LP-filter consists of a one-pole LP-filter in series with a non-inverting OP-amp, which can mathematically in the frequency domain be written as

$$\begin{aligned} V_{out2} &= \left(\frac{2\pi f_0}{j2\pi f + 2\pi f_0} \right) \left(1 + \frac{R_{10}}{R_9} \right) V_{out1} \\ &= \left(\frac{1}{1 + j \frac{f}{f_0}} \right) \left(1 + \frac{R_{10}}{R_9} \right) V_{out1} [\text{V}], \end{aligned} \quad (3)$$

where the cutoff frequency is

$$f_0 = \frac{1}{2\pi R_8 C_1} [\text{Hz}]. \quad (4)$$

The purpose of the A-LPF is to filter out the AC disturbances since the system only handles DC signals and amplifies the final output from the ACC.

4) *ACC-summary*: By combining equations (1), (2) and (3) a transfer function between the output from the WB and the output V_{out2} from the ACC can in frequency domain be written as

$$\begin{aligned} V_{out2} &= \left(\frac{1}{1 + j \frac{f}{f_0}} \right) \left(1 + \frac{R_{10}}{R_9} \right) \frac{R_7}{R_6} \left(1 + 2 \frac{R_5}{R_4} \right) \\ &\quad \cdot E \left(\frac{R_3}{R_1 + R_3} - \frac{R_s}{R_2 + R_s} \right) \\ \{A_{LP} &= \frac{1}{1 + j \frac{f}{f_0}} \left(1 + \frac{R_{10}}{R_9} \right) \} \\ \{A_{IA} &= \frac{R_7}{R_6} \left(1 + 2 \frac{R_5}{R_4} \right) \} \\ V_{out2} &= A_{LP} \cdot A_{IA} \cdot E \left(\frac{R_3}{R_1 + R_3} - \frac{R_s}{R_2 + R_s} \right) [\text{V}]. \end{aligned} \quad (5)$$

The parameters A_{LP} and A_{IA} are introduced to make the transfer function more compact and to give a clearer picture

of how the output from the bridge changes the output for the entire ACC. The parameter A_{IA} is a constant amplification factor that comes from the IA. The parameter A_{LP} represents the A-LPF, where the amplification can be regulated by the variable resistor R_{10} . From equation (5) the following can be concluded:

- Since the ADC, more details under section III-H MCU, can not read negative voltage values, the range for the magnitude $|V_{out2}|$ that can be measured by the ADC, assuming R_s can equal 0Ω will lie between the following values

$$0 < |V_{out2}| < |A_{LP} A_{IA} E \frac{R_3}{R_1 + R_3}| [\text{V}].$$

- For high frequency disturbances the magnitude of V_{out2} will tend towards zero, since

$$f \rightarrow \infty \implies |A_{LP}| \rightarrow 0 \implies |V_{out2}| \rightarrow 0.$$

- The total amplification denoted G of the ACC lies between

$$1 < G < \left(1 + \frac{R_{10}}{R_9} \right) \left(1 + 2 \frac{R_5}{R_4} \right)$$

assuming R_5 and R_{10} can equal 0Ω .

B. MCU

The MCU have the following functionality:

- Measuring the magnitude of the output signal $|V_{out2}|$ from the ACC, which is done via an internal ADC channel that converts $|V_{out2}|$ to a digital signal $V_{out2}[n] = |V_{out2}(nT_s)|$ via sampling (where n is the sample number and T_s the sampling period) and sends $V_{out2}[n]$ to a computer.
- Adjusting the WB via R_3 and the gain in A_{LP} via R_{10} .

In this implementation R_3 and R_{10} are digital potentiometers that follows the I²C communication protocol, more details under section II-C. The MCU can step-wise increment, decrement, or set a specific value for R_3 and R_{10} in software by sending data packets that follow this protocol. A block diagram of the ACC combined with the MCU can be seen in figure 3.

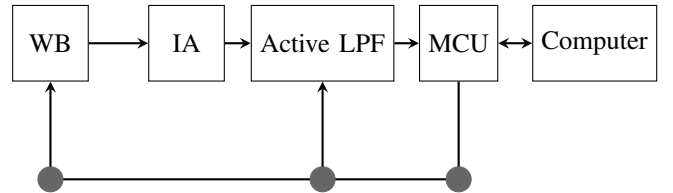


Fig. 3. Block diagram of the DAU. The sensor is connected in the WB. The voltage signal from the sensor goes through the IA and A-LPF to the MCU. The MCU measures the signal and sends it to a computer for display. The MCU controls R_3 in the WB and R_{10} in the A-LPF through I²C.

C. I²C

I²C is a serial communication protocol that uses two wires for communication with one master device (in this case the MCU) and one or multiple slave devices (in this case R_3 and R_{10}), this can be seen in figure 4. The communication is done

with two wires called SDA and SCL, where the SDA wire is used to send and retrieve data and the SCL wire is used to carry the clock signal from the main device. SDA and SCL are both connected via two pull-up resistors denoted R_p to a high reference voltage denoted V_{cc} . SDA and SCL can only equal the voltage value 0.0 V or V_{cc} [V]. The voltage 0.0 V represents the binary value zero and is called LOW, whereas V_{cc} [V] represents the binary value one called HIGH. The reference voltage V_{cc} usually equals 3.3 V or 5 V [6]. In order for the main device to write to a slave device it needs to execute the following sequence in the enumerated order:

- 1) **Start condition:** This condition tells the bus that the master device wants to start communicating. The condition is that the master device pulls SCL HIGH during a fixed time interval simultaneously pulling SDA from HIGH to LOW.
- 2) **Address byte in write mode:** After the start condition the master device sends a byte, where the first 7 bits contain the address to the slave device of interest on the bus and the last bit equals one indicating that the device shall be written to. The master device will then wait for an acknowledgment (ACK) bit from the slave in order to continue the sequence.
- 3) **Register address byte:** When an ACK bit has been received from the selected slave device the master will send a byte containing the memory location inside the slave that it wants to write data to. After this, the master will wait for an ACK bit from the slave before proceeding in the sequence.
- 4) **Data byte:** This byte contains the data the master device wants to send to the memory location given by the register address from the previous byte. After this byte has been sent a final ACK bit will be sent from the slave to the master confirming that the data was received.
- 5) **Stop condition:** In order to finish the sequence a stop condition needs to be fulfilled by the master device. This condition tells the selected slave device that the transmission is done and is identical to the start condition with the only difference being that SDA is pulled from LOW to HIGH instead of HIGH to LOW.

In order for the master device to read data from a slave device the following sequence in the enumerated order needs to be executed:

- 1) **Start condition:** Identical to the write sequence.
- 2) **Address byte in read mode:** Identical to the write sequence with the only difference being the last bit equaling zero, indicating to the slave that the master wants to read from it. After this byte is sent the master will wait for an ACK bit from the slave.
- 3) **Register address byte:** This byte is identical to byte in the write sequence. After this byte is sent the slave shall transmit the data stored in the memory location given by the register address to the master. The master will then send an inverted ACK bit called NACK to the slave indicating that the data was received.
- 4) **Stop condition:** Identical to the write sequence.

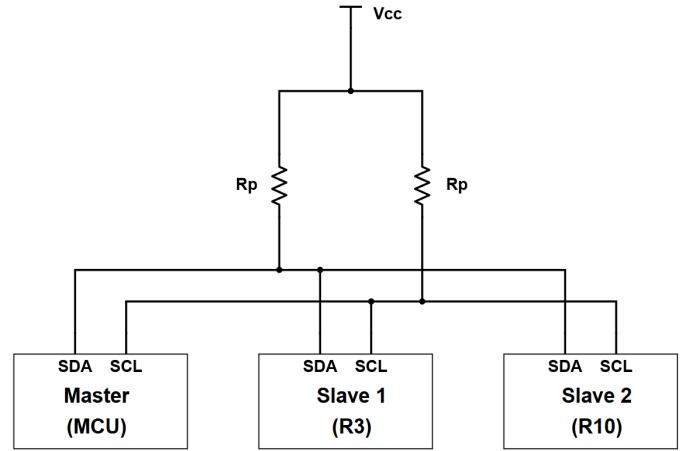


Fig. 4. A diagram showing an example of how the MCU, R_3 and R_{10} could be connected according to the I²C protocol.

III. IMPLEMENTATION

A. Calibration of a sensor

Reading sensible data from sensors requires calibrations. The calibration is a necessary step in order to match the output of the system to a specific input of the physical parameter the sensor measures. A DAU allows for measurement in a certain range. Commonly the range is specified for the system and is limited by the actual components implemented in the ACC.

The calibration starts by deciding the zero output, which is done by measuring the output when the input is considered zero of a physical parameter [7]. This could for example be when a temperature sensor measures 0°C. The measured input to the DAU should be within the calibrated range to fit within the range of the electric output of the DAU, in this case, 0.0 V to 3.3 V. The next step is to do a sequence of calibrations until reaching the maximum value of the sensor's full-scale output. In case the sensor's resistive value can exceed 50 kΩ, which is the highest resistance the DAU can handle (see III-E), the calibration can only be done for the range of 0 kΩ to 50 kΩ. This limits the types of sensors that can be used and the range in which the sensor can be used in.

The calibration requires a number of calibration points, the amount of points varies from sensor to sensor. With linear sensors it can be sufficient to have three points of calibration; a zero input, a full-scale input, and zero input again. However, for non-linear sensors, it is recommended to have more calibration points, an eleven-point calibration for example, with a 20% interval between them, from the zero input to the full input and back to the zero input. In the case of a sensor that has two directions, above and below the defined zero input of the sensor, it is recommended to have additional calibration points in the negative direction. More calibration points can be added if necessary. It is also important to repeat the calibration cycles under well-defined test circumstances. A certain confidence interval can be obtained and is often decided by the measurement needs and usage area of the sensor. To provide meaningful data, the calibration system's cumulative errors or uncertainties must be less than the specified tolerance of the performance parameter

under evaluation. Different methods can be used to find an equation that fits the data points such as linear or polynomial regression [7].

External measuring equipment is required to compare the measured value of the system with an ideal value of the measurement. The uncertainty of the equipment must be less than the uncertainties from the system, otherwise, the result is insignificant. If the deviation or error is bigger than the allowed deviation specified by the sensors data-sheet a re-calibration must be done.

B. Calibration of the DAU

The calibration is done by adjusting the digital potentiometers of the system by the MCU. This must be executed manually for each sensor in the software for the MCU. It is very important to maintain a well-defined condition under the calibration process. It has three important steps:

- Calibration of the digital potentiometer in the WB
A sequence of calibration needs to be done to adjust the measured input of the sensor to desired output value by controlling the resistance value of the digital potentiometer R_3 . The DAU allows for readjustment of the desired output for a certain range of the sensors if it might be beneficial to decrease uncertainties from the measured data. It can be done by readjusting the values of the digital potentiometers in the WB and A-LPF.
- Adjusting the gain of the A-LPF
The adjustment of the WB is discrete because there are limited values of the digital potentiometer R_3 that can be selected. In the second step, the readjustment of the gain by the OP-amp U_4 in the A-LPF by changing the digital potentiometer R_{10} can be done to minimize the deviation from the desired output value. A linear system is always desired since it minimizes the need for recalculation of the measured value.
- Saving calibration values
After the calibration is performed and the desired values for the digital potentiometers are obtained, it is necessary to save them in order for the system to use the same calibration settings while running the system.

C. Hardware

Components for the hardware platform have been chosen to fit the design limitation for the ACC that can work together with an available MCU. The components have been selected after availability in stocks and from the standard components library of the KTHFS team. The available MCU for this thesis was a STM32F0-series chip, more details regarding the chip are explained in III-H. Most of the components that are used in the DAU work with a DC supply of 5 V and the microcontroller only works for 3.3 V, which meant that the supply of 24 V needed to be stepped down to both 5 V and 3.3 V. This was done by using two regulators, one that transformed 24 V to 5 V and one that transformed 5 V to 3.3 V. The final prototype was done on a PCB where details are described in III-L. The block diagram used for the implementation of the DAU can be seen in figure 5.

D. Regulators

The regulator used to step 24 V to 5.0 V was a 173950x78 regulator created by WURTH ELEKTRONIK [8]. This regulator can handle input voltages between 6.0 V to 24 V and output a fixed voltage of either 3.3 V or 5.0 V. The other regulator that stepped down 5.0 V to 3.3 V was an ADP150 from Analog Devices [9]. This regulator can handle input voltages between 5.5 V to 2.2 V and output a fixed voltage of either 3.3 V or 1.8 V, to minimize ripple [9] recommends one $1 \mu\text{F}$ capacitor on the input and output of the device.

E. WB

The supply voltage for the WB was set to $E = 5.0 \text{ V}$, the upper leg resistors R_1 were selected as $100 \text{ k}\Omega$ resistors and the digital potentiometer MCP4541 manufactured by MICROCHIP [10] was used as R_3 . Its maximum value is $50 \text{ k}\Omega$ and it contains a resistor network of 128 resistors, giving it a resolution of

$$\frac{50 \text{ k}\Omega}{128 \text{ steps}} \approx 390 \frac{\Omega}{\text{step}}.$$

The reason for choosing a digital potentiometer with this resolution was for the most part due to component accessibility where this was the best resolution that was found for the maximum resistance of $50 \text{ k}\Omega$. Since this limits the possibility to adjust the output of the WB for calibration an adjustable gain has been added to expand the flexibility and increase the adjustment during calibration of a sensor, see under section III-G A-LPF. The maximum value of $50 \text{ k}\Omega$ limits the dynamic range of the resistance R_s for the connected sensor WB. A sensor with a value bigger than $50 \text{ k}\Omega$ will result in a negative voltage output, which can be seen using the equation (1). The value of $50 \text{ k}\Omega$ has been chosen since most of the resistive sensors connected to the vehicle are in the range of $50 \text{ k}\Omega$. Negative output voltages from the WB will be saturated by the OP-amp U_4 in the A-LPF, since it is being single supplied with the negative rail being ground (0 V). The MCP4541 requires a supply voltage between 1.8 V to 5 V. It has 8 pins, a high and low potentiometer terminal, a wiper terminal that can switch between different values of the resistor, and one address pin used to create the device address allowing for two unique addresses, meaning that only two MCP4541 can be used on the same I²C communication bus. The MCP4541 has low wiper resistance, around 75Ω and the total resistance of the MCP4541 can be modeled as:

$$R_d = 390n + R_w [\Omega] = 390n \Omega + 75 \Omega, \quad n = 0, 1, \dots, 128,$$

where R_d is the resistance of the digital potentiometer, R_w is the resistance of the wiper and n the number of steps.

F. IA

The IA implemented in the circuit was a INA114AU created by Burr Brown [11]. It has low offset voltage of $50 \mu\text{V}$, low input bias current of 2 nA and a high CMRR up to 115 dB and offers supply range as low as $\pm 2.25 \text{ V}$. The CM input range

should be 1.25 V from the supply range. The gain of for the INA114AU is given by

$$A_{IA} = 1 + \frac{50 \text{ k}\Omega}{R_4}.$$

In this implementation $R_4 = 1.5 \text{ M}\Omega$ which implies that $A_{IA} \approx 1$ (unity gain). The purpose of the IA is to impedance match the differential signal from the WB and to utilize the high CMRR, leaving the amplification to the A-LPF.

G. A-LPF

The implemented LPF consists of a constant resistance with a value of $R_8 = 10 \text{ k}\Omega$ and a constant capacitor with the value of $C_1 = 1.0 \mu\text{F}$, giving a cutoff frequency according to equation (4) of:

$$f_0 = \frac{1}{2\pi \cdot 10 \text{ k}\Omega \cdot 1.0 \mu\text{F}} \approx 16 \text{ Hz}.$$

The non-inverting OP-amp U_4 was a TLV9061 designed by Texas Instruments [12]. It offers a low input offset voltage of $\pm 0.3 \text{ mV}$, a low input bias current of 0.5 pA and most importantly it is designed for low-voltage operation where the rail to rail difference voltage can lie between 1.8 V to 5.5 V. In the non-inverting amplifier circuit, a MCP4541 was used as R_{10} and a constant resistor of $10 \text{ k}\Omega$ was used as R_9 , which according to equation (3), gives a total gain that can lie between 1-6 times depending on the value of R_{10} . The gain can be calculated as following:

$$G = 1 + \frac{R_{10}}{R_9} = 1 + \frac{390n \Omega + 75 \Omega}{10 \text{ k}\Omega}, \quad n = 0, 1, \dots, 128.$$

The adjustable gain adds to the flexibility of the system and has an essential role in calibrating the sensors, so the input of the sensor matches a desired output. Furthermore, the OP-amp U_4 is supplied with 3.3 V to 0.0 V and therefore matches the output of the IA to the input of the ADC. A signal from the IA which is greater than 3.3 V or lower than 0.0 V will saturate ensuring that the final output $|V_{out2}|$ from OP-amp U_4 lies in the range $[0.0 \text{ V}, 3.3 \text{ V}]$. Also, the resistance value of $R_8 = 10 \text{ k}\Omega$ ensures that the current into the OP-amp U_4 never exceeds 10 mA , which is a requirement for it to function properly according to the datasheet [12].

H. MCU

The MCU implemented in the circuit is a STM32F091RC created by STMicroelectronics that contains a high-performance ARM-Cortex-M0 chip [13]. It offers a wide range of enhanced peripherals and standard communication interfaces such as I²C (two channels), CAN, and a 12-bit ADC which are necessary to control and measure the ACC and to communicate with a computer. The internal ADC can sample between $1 \mu\text{s}$ to $17 \mu\text{s}$ and has a input voltage range of 0.0 V to 3.3 V. It also has an ADC oversampling feature that makes it possible to average each measurement, in order to obtain a more accurate value.

I. I²C bus

The MCU is connected to each digital potentiometer R_3 and R_{10} via two separate I²C channels called I²C₁ and I²C₂. The pull-up resistors for each channel were chosen as $R_p = 1 \text{ k}\Omega$ and the reference voltage as $V_{cc} = 3.3 \text{ V}$ by being connected to the 3.3 V voltage supply.

J. JTAG and ST-Link

The mounted JTAG (Joint Target Action Group) connector is a 10 pin male connector that is used together with a ST-Link (debugger probe) for communication purposes between the STM-chip and a computer. The communication is done through the Serial Wire Debug (SWD) interface, which is a serial wire protocol that uses two pins called SWDIO and SWDCLK. It can be used to program and read data from the chip memory [13].

K. Overview of the system

An overview of how the different parts of the system are connected can be seen in figure 5. The blocks with voltage values represent the power supply in the DAU and the arrows between them a step-down transformation. The WB and IA are supplied with 5.0 V and the A-LPF and MCU with 3.3 V. The arrows pointing to the right between the ACC (WB, IA and A-LPF) and MCU represent the voltage signal that carries the value of the sensor. Which in turn the MCU sends through the JTAG and ST-Link to a computer. Code can also be uploaded to the MCU from a computer via the JTAG and ST-Link. The arrows from the MCU to the WB and A-LPF represent the I²C communication to the digital potentiometers R_3 (WB) and R_{10} (A-LPF). The optional CAN block represents the connections on the PCB that can be used to mount components required for CAN communication, however, the software has not been developed for it.

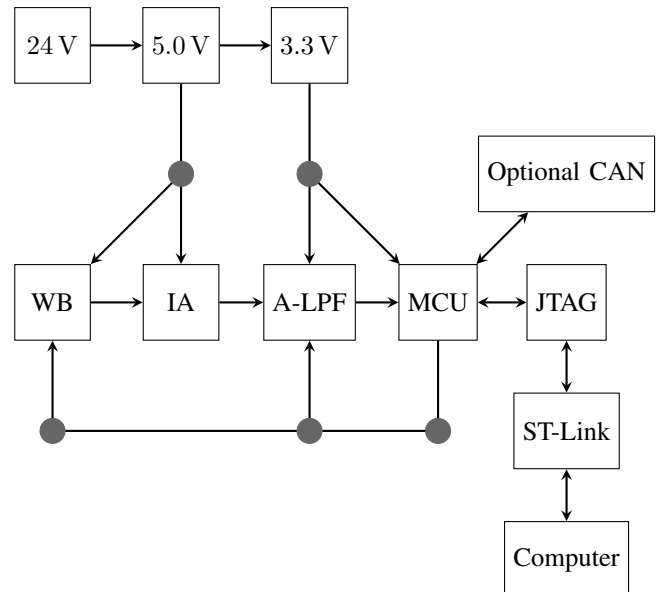


Fig. 5. Block diagram illustrating how the implemented blocks are interconnected in the DAU.

Adding all the parts of the DAU with their given parameter values the total system specification can be expressed as table I.

TABLE I
SYSTEM SPECIFICATIONS FOR THE PROPOSED DAU

PARAMETER	Values
Supply Voltage	24 V
Output Voltage Range	0 V - 3.3 V
Input Resistance Value	0 Ω - 50 k Ω
G (Gain factor)	1 - 6
Resolution of the ADC	1 mV
Sampling time of the ADC	1 μ s - 17 μ s
Cutoff frequency f_0	16 Hz

L. PCB design of the hardware

The two layered PCB was designed using KICAD, the render of the circuit 3-D model is shown in figure 6. Certain design rules were followed to optimize the board construction and minimize errors. The PCB was printed using a LPKF milling machine, located at KTH Mentorspace. Therefore there were certain design rules and guidelines that had to be followed to match the machine requirements such as sizes of vias and traces.

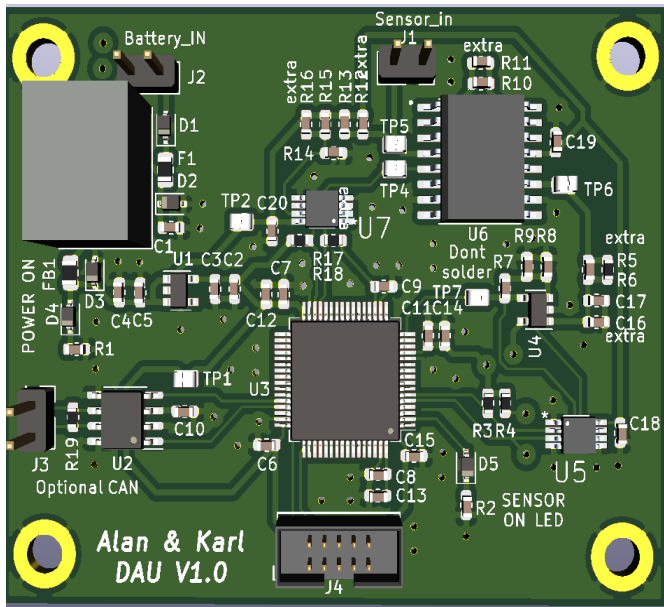


FIG. 6. The 3-D model of the designed PCB for the DAU. It is designed using KICAD v5.0.

M. Software for the MCU

The MCU was programmed in the software tool STM32CubeIDE created by STMicroelectronics, which supports the programming languages C and C++. The code that was used in the MCU was written in the IDE on a computer and could be uploaded via the ST-Link and JTAG connector with the SWD interface. The code utilizes two functions called: 1. `digi_adjust` and 2. `ADC_read`. The implementation of these functions utilizes predefined functions

from the STM32F0 HAL library, further details regarding them can be found in [14]. The functions used from the library were:

- `HAL_I2C_Master_Transmit`: Takes the 7-bit device address and sets the LSB (Least Significant Bit) to one when the device address is left shifted one time. Then sends the user-defined data bytes in accordance with I²C via a user-selected channel.
- `HAL_I2C_Master_Receive`: Similar to the transmit function takes the 7-bit device address and sets the LSB to zero when the device address is left shifted one time. Then stores the received data from the register inside the slave device into a declared variable via a pointer.
- `HAL_ADC_Start`: Enables a user selected ADC inside the MCU making it sample the analog signal and converting it to a digital value.
- `HAL_PollForConversion`: Halts the program until the conversion of the analog signal to a digital value is complete.
- `HAL_ADC_GetValue`: Returns the digital value from the sampling of the analog signal.
- `HAL_ADC_Stop`: Disables the ADC.

The function `digi_adjust` always reads the current wiper value of a digital potentiometer and can increment, decrement that value or set a specific value for the wiper. In order to achieve this with the MCP4541 model the following bytes in the code needed to be defined in the function in hexadecimal form:

- **Address byte:** 0x2E is the 7-bit address for MCP4541 since the pin A0 on the device is connected to the ground, which can be seen at [p.50, 10] and the LSB is either one for write or zero for read. Since the digital potentiometers R_3 and R_{10} were put on two different I²C channels only one address byte had to be defined because they can use the same address.
- **Register address byte:** 0x00, which is the register address for the wiper inside the MCP4541 (see Volatile Wiper 0 at [p.56, 10]).
- **Data byte:** 0xNN, NN needs to be chosen in such a way that the entire byte in decimal form lies in the integer interval 0 to 128 (0x00 to 0x80 in hex), since the wiper of a MCP4541 only has 128 steps. This byte is only used when the wiper position shall be changed.

The pseudocode for `digi_adjust` can be seen in **Algorithm 1**. The function `ADC_read` stores the current digital value of a user-selected ADC channel in a variable and returns it. The pseudocode for `ADC_read` can be seen in **Algorithm 2**.

N. Perfboard

Due to problems with the PCB it was not possible to do any testing with it, the reasons for this are explained in IV-A. Because of this a prototype of the ACC was built on a perfboard, see figure 7. This was built due to time shortage and limited access to rail-to-rail input-output instrumentation amplifiers therefore dual supplies were required to operate the circuit in full range.

Algorithm 1 Pseudocode for **digi_adjust****digi_adjust(sel, data, channel):**

- 1: **declare** variable *W_val* //Variable that stores the wiper position.
- 2: **declare** variable *address* = 0x2E //MCP4541 device address.
- 3: **declare** array *data*[2] //data[0]: Register address byte, data[1]: Data byte.
- 4: *data*[0] = 0x00 //Sets *data*[0] to the MCP4541 wiper register address.
- 5: **HAL_I2C_Master_Transmit**(channel, *addressL*, *data*[0])
//Sends register byte and *addressL* is a dummy notation for one bit left shifting of the device address.
- 6: **HAL_I2C_Master_Receive**(channel, *addressL*, **W_val*)
//Receives and stores the wiper value in *W_val* with the pointer **W_val*.
- 7: **if** (*sel* = 0) **then**
- 8: *data*[1] = *W_val* - 0x01 //Sets the data byte equal to the received wiper value decremented with one step.
- 9: **else if** (*sel* = 1) **then**
- 10: *data*[1] = *W_val* + 0x01 //Sets the data byte equal to the received wiper value incremented with one step.
- 11: **else if** (*sel* = 3) **then**
- 12: *data*[1] = *data* //Sets the data byte equal to the inputted data byte.
- 13: **end if**
- 14: **HAL_I2C_Master_Transmit**(channel, *addressL*, *data*[2])
//Sends the two bytes *data*[0] and *data*[1] to the digital potentiometer.

Algorithm 2 Pseudocode for **ADC_read****ADC_read():**

- 1: **declare** variable *ADC_val* //Variable where the ADC value shall be stored.
- 2: **HAL_ADC_Start**
- 3: **HAL_ADC_PollForConversion**
- 4: *ADC_val* = **HAL_ADC_GetValue** //Storing the ADC value in *ADC_val*.
- 5: **HAL_ADC_Stop**
- 6: **return** *ADC_val* //Returns the variable *ADC_val*

O. Testing setup with perfboard

The ACC circuit on the perfboard was connected to a microcontroller board containing the STM32F091RC chip with the necessary connectors I²C, ADC channel, and a debugger interface via USB to a computer. The STM32CUBEIDE software was used to upload code to the MCU on the microcontroller board from the computer. The code allowed via the IDE to read from and control the ACC. The testing was done by calibrating a linear displacement sensor called SLS095, which can measure displacement between 10-100 mm. It has a resistance between $400 - 4000 \Omega \pm 10\%$, check datasheet [15]. Two RND LAB DC power supplies 320-kd3005d were used to supply ± 5.0 V to the perfboard DAU. A measurement tape was used to measure the displacement of the sensor. The test was done at room temperature and at normal

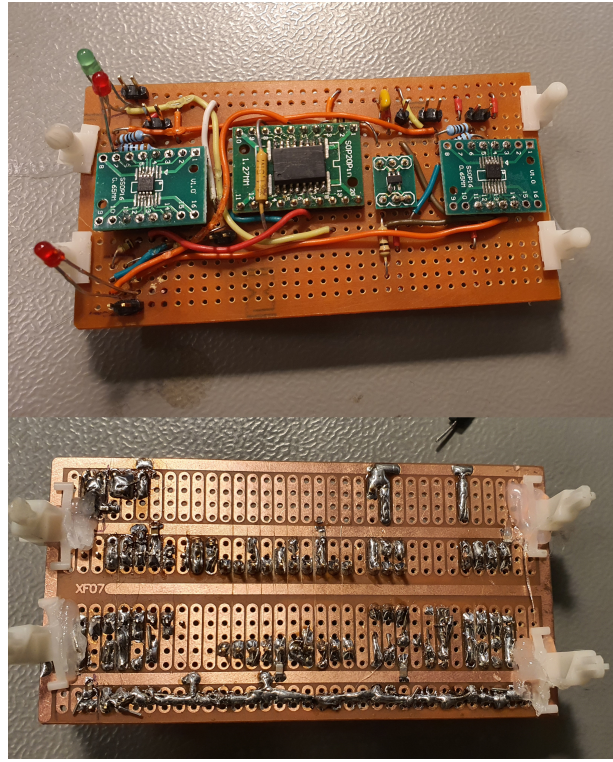


Fig. 7. The build ACC on a perfboard with connectors going to a microcontroller board and supply bins going to power source.

humidity. In total, 6 sets of measurements were done under the same conditions. Each set measured one of 11 discrete displacements for the sensor with a 10 mm difference between each displacement. To set a reference voltage of 0.0 V when the displacement was 0.0 mm the functions *digi_adjust* and *ADC_read* were used in a while-statement in the code for the MCU. This was done by first setting the digital potentiometers R_3 and R_{10} to their max value (wiper position 128) with *digi_adjust*. Afterwards *digi_adjust* and *ADC_read* were used inside the while-statement, where the condition for the statement was to run until the read ADC value from *ADC_read* reached the reference voltage when simultaneously step-wise decrementing R_3 and R_{10} . The results of this test can be seen in IV-B.

IV. RESULT

A. The assembled PCB

The assembled PCB is shown in figure 8. The first prototype has additional pads for resistors and capacitors in case the values need to be readjusted. The CAN components were not assembled, but they can be soldered for future testing. The MCU was successfully mounted and tested on the PCB, but the PCB couldn't be used for testing due to a wrong connection on the reference pin for the IA. The reference pins were connected to the ground instead of being connected to half the supply voltage, in this case, 2.5 V. A way of implementing a reference voltage by 2.5 V is by adding a voltage divider combined with a voltage follower. The INA114AU doesn't operate properly on the voltage that's less than 1.25 V from supplies and therefore it will saturate values within the range

of the ADC which might be necessary for sensor calibration. Therefore a rail-to-rail input output instrumentation amplifier is more desired for such application. Another problem with PCB is the footprint for the TLV9061 was wrong. There is two different packages with the same footprint but different orientation of the output and input voltage of the OP-amps. Due to the lack of another microcontroller and time shortage, a perfboard was built and tested.

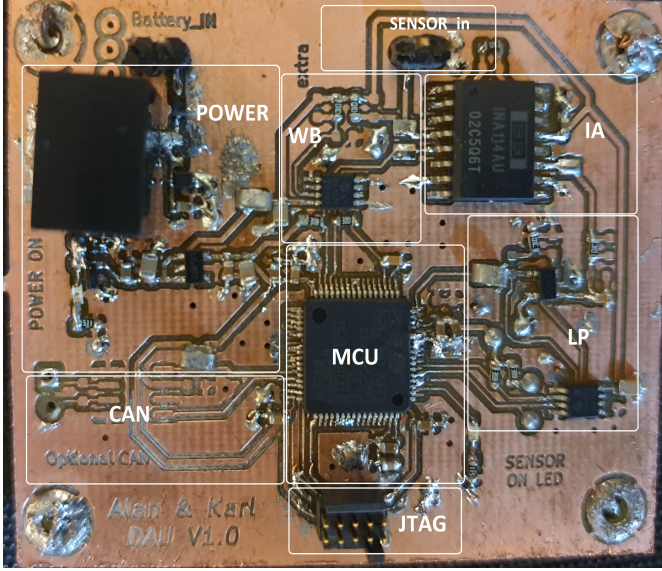


Fig. 8. The assembled PCB. The upper layer is a copper layer. VIAS have a copper wire soldered through them

B. Experimental results

The first step is the calibration of the sensors. The calibration can be used as a manual to translate the voltage value to a physical parameter. The result of the measurement with the SLS095 sensor is presented in table II, where the parameter D is the displacement of the sensor and $x_{i=1,\dots,6}$ is the output voltage for a given displacement D . From table II a linear regression plot was made where the mean of each data set were plotted against the corresponding displacement, which is presented in figure 9. Assuming a normal distribution for the measurements, 95% of all measurements are between the mean value and two standard deviation values, written as $\mu \pm 2\sigma$.

Linear regression was used to calculate a linear equation that fits the measurement points. This equation can be used while using the sensor to calculate the displacement from measured voltage using the ADC channel in the MCU and displayed on a computer. To evaluate the calibration, a test has been conducted in the same environment to test the accuracy of the system, and if the calibration model is sufficient enough to be used. The results are shown in table III. The result shows that all voltage measurements were within the confidence interval for the different displacements. The maximum error in displacement was 3.1 mm.

V. DISCUSSION

A DAU has the potential to minimize the hardware needed for sensor measurement by having an adjustable digital com-

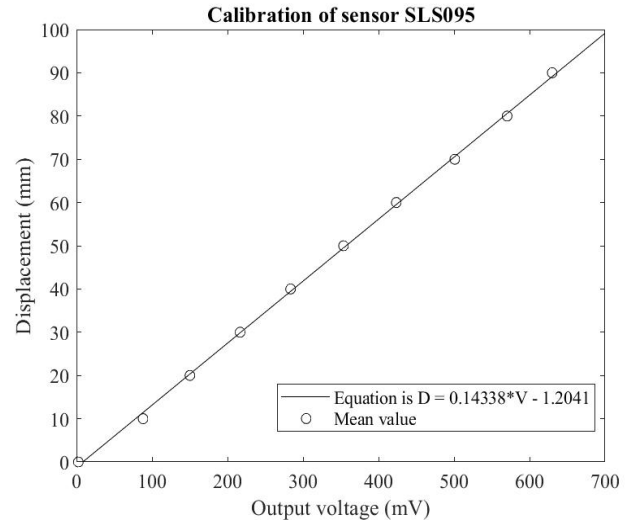


Fig. 9. The mean values of the 6 different tests have been used to find a line that fits the measurement. The mean values align well within the line due to the sensor being linear.

ponent controlled by software. It provides a flexible design and can be used to calibrate a variety of sensors.

A. Limitations of the system

A measurement instrument device is specified for a range of measurements. This range is determined by the limit of the system. The system's limitations are divided into two categories; predefined by the problem and by design choice. The current design choices are done to test the concept of digital adjustable ACC and can be readjusted for future DAU to fit the area of measurement, for example a higher range for resistive sensors. Starting with supply voltage, the system can work for voltages between 6.0 V to 28 V. The value of the resistive sensor R_s should not exceed the values of the digital potentiometer R_3 , implying that the resistance range for the sensor must lie between 0–50 k Ω . The CM input range of the IA should be 1.25 V from the supply voltage according to its datasheet. This limits the range of the input to 1.25–3.75 V. Therefore a rail-to-rail input output INA is more suitable for this application. The OP-amp TLV9061 has a full rail-to-rail input-output range allowing for input between 0–3.3 V which is good since that is the entire range of the used ADC.

The STM32F091RC is a powerful chip, that has a high sampling rate on its ADC. The minimum sampling time is 1 μ s, allowing for up to 1 million samples per second (MSPS), the sampling rate is controlled by the internal clock of the MCU. This sampling rate is sufficient enough for most sensor systems used in the vehicle and allows for measurements from additional sensors without a major delay. The limitation is by the ADC-supply range from 2.4–3.6 V. This in turn limits the measurement range to 0–3.3 V for the output V_{out2} of the ACC. Furthermore, the ADC has a 12-bit resolution which combined with the supply voltage of 3.3 V gives the smallest readable voltage as

$$\frac{3.3 \text{ V}}{2^{12}} = \frac{3.3 \text{ V}}{4096} \approx 1 \text{ mV}.$$

TABLE II
RESULT OF MEASUREMENT WITH SLS095

Displacement D: [mm]	0.0	10	20	30	40	50	60	70	80	90	100
x_1 [mV]	0	90	149	203	277	344	413	485	553	610	684
x_2 [mV]	5	84	143	205	273	351	413	487	570	610	705
x_3 [mV]	10	90	151	217	285	352	428	505	555	620	684
x_4 [mV]	0	89	160	230	291	363	435	510	581	655	730
x_5 [mV]	0	88	150	231	288	359	425	510	580	637	725
x_6 [mV]	0	88	150	231	288	359	425	509	583	648	725
Mean(column): μ [mV]	2.50	88.2	150	217	284	354	424	501	570	630	708
Standard Deviation: σ [mV]	4.18	2.23	5.56	11.9	6.98	6.62	8.80	11.8	13.4	19.5	21.1

TABLE III
TESTING OF THE CALIBRATION

Real displacement [mm]	Measured voltage [mV]	Calculated displacement [mm]	Error [mm]
0	0	-1.2	-1.2
10	88	11.4	1.4
20	157	21.3	1.3
30	230	31.8	1.8
40	293	40.8	0.8
50	362	50.7	0.7
60	436	61.3	1.3
70	510	71.9	1.9
80	588	83.1	3.1
90	655	92.7	2.7
100	730	103	3.0

Therefore the highest resolution the DAU can achieve is 1 mV. Achieving a higher resolution and voltage range requires another ADC.

B. Testing with the PCB

As mentioned in IV-A the assembled PCB could not be used for testing. Lacking time and also access to another MCU it was not possible to make a new fully functioning PCB. Some further difficulties that weren't mentioned regarding the assembling process in IV-A were:

- The lack of silk screen made it harder to identify where the different components should be placed.
- Non-plated through holes for vias connections which required that a copper wire was soldered on each layer of the PCB that went through the hole.
- Small pads for components making it easy to short them together.

Regarding testing with a functioning PCB it's expected that it should perform better than the perfboard variant. A PCB is less noisy compared to a perfboard since all components are placed near each other without the need for cables, which minimizes noise from parasitic elements from them. Also decoupling capacitors can be placed closer to all active components in the circuit, which both stabilizes their supply and reduces noise.

C. Testing with different sensors

Unfortunately due to a lack of testing instruments and shortage of time, the system has not been verified to work with more than one sensor being the SLS095 displacement sensor. The test demonstrates that it is possible to calibrate a

resistive sensor that has a resistance span of 400 – 4000 Ω into a specific voltage range. Testing with different sensors with smaller or bigger resistive ranges is needed to determine the systems's capability to work with more sensors.

D. Insufficient characterization of the system

The current design has the capabilities to filter AC-noise. Therefore testing the effect that AC noise has on the system is needed in order to calculate the SNR (Signal to Noise Ratio) of the system. It can be achieved by introducing an AC signal into the sensor input or the supplies of the system. In case the cutoff frequency is too high the values of R_8 and C_1 need to be changed. The used IA has a high CMRR as stated in the datasheet. It should be tested in the DAU to specify how well it performs in practice. Lastly measuring the dynamic range (DR) of the system to quantify the ratio between the maximum and minimum detectable value. Due to lack of accurate measurement equipment and time shortage, these three parameters haven't been measured.

E. Calibration

The current design requires prior knowledge of environmental noise factors such as temperature. Therefore incorporating a way to automatically compensate for the offset produced from the environment the DAU is in could be of great importance. One way of doing this could for example be by adding an additional hiding sensor in the WB. The measurement from the hiding sensor could compensate for the offset and through analog conditioning solve the issue. Another option could be adding an offset trimming circuit for the output offset voltage of the IA. The current design allows for offset adjustment if the usage conditions are previously recognized and can be accounted for under calibration.

F. Future work

Being the first prototype of a DAU there are more aspects that could be investigated in the future. Adding additional ACC circuits that can handle AC signals for the measurement of capacitive and inductive sensors. If the DAU shall be placed inside a formula vehicle where CAN bus is being used to communicate with other subsystems a software implementation needs to be implemented for it. Testing a fully functioning PCB of the system in a harsh environment that the vehicle could be driven in to see how well it performs with respect to noise, precision and accuracy.

VI. CONCLUSION

The first prototype of DAU design provides the benefit of digitally controlling the hardware for sensor calibration and measurement. Further testing is required to specify the effect of noise, precision, and accuracy on the system also demonstrating its capabilities to work for different sensors. For the DAU to be able to communicate to other subsystems in the vehicle a way to implement CAN communication needs to be investigated.

ACKNOWLEDGMENT

The authors would like to thank Mark T Smith for supervising the project and helping with the designing and manufacturing of the system. We are also grateful for the ideas contributed by Carl-Mikael Zetterling and Matthias Becker. Thanks also to KTH Formula Student for providing assistance and resources during the development of the system.

REFERENCES

- [1] A. De Marcellis, C. Reig, and M.-D. Cubells, "A novel current-based approach for very low variation detection of resistive sensors in wheatstone bridge configuration," in *SENSORS, 2014 IEEE*, Dec. 2014, pp. 2104–2106.
- [2] F. Barišić, K. Špoljarić, H. Hegeduš, and P. Mostarac, "High precision data acquisition system for resistance measurement with wheatstone bridge," in *2020 3rd International Colloquium on Intelligent Grid Metrology (SMAGRIMET)*, Nov. 2020, pp. 104–108.
- [3] M. Di Paolo Emilio, *Embedded Systems Design for High-Speed Data Acquisition and Control*. Switzerland: Springer, 2015, p. 131.
- [4] S. Corrigan. (2016, May) Introduction to the controller area network (can). Texas Instrument, Dallas, TX, USA. [Online]. Available: <https://www.ti.com/lit/an/sloa101b/sloa101b.pdf>
- [5] C. Kitchin and L. Counts, *A Designer's Guide to Instrumentation Amplifiers*, 3rd ed. USA: Analog Devices, Inc., 2006, ch. 2, pp. 2–6.
- [6] J. Valdez and J. Becker. (2015, Jun.) Understanding the i^2c bus. Texas Instrument, Dallas, TX, USA. [Online]. Available: <https://www.ti.com/lit/an/slva704/slva704.pdf>
- [7] P. S. Lederer, *Sensor Handbook for Automatic Test, Monitoring, Diagnostic, and Control Systems Applications to Military Vehicles and Machinery*. Commerce Department, National Institute of Standards and Technology (NIST), 1981, no. 615, pp. 151–157.
- [8] Würth Elektronik eiSos GmbH Co. (2019, May) 173950x78 magi³c power module fdsd - fixed step down regulator module. Würth Elektronik, Waldenburg, Germany. [Rev. 2.0]. [Online]. Available: <https://www.we-online.com/catalog/datasheet/173950578.pdf>
- [9] Analog Devices. (2020, Aug.) Ultralow noise, 150 ma cmos linear regulator. Analog Devices, Norwood, MA, USA. [Rev. D]. [Online]. Available: <https://www.analog.com/media/en/technical-documentation/data-sheets/ADP150.pdf>
- [10] Microchip Technology. (2013, Feb.) 7/8-bit single/dual i^2c digital pot with nonvolatile memory. Microchip Technology, Chandler, AZ, USA. [Rev. B]. [Online]. Available: <https://ww1.microchip.com/downloads/en/DeviceDoc/22107B.pdf>
- [11] Burr Brown. (1998, Mar.) Precision instrumentation amplifier. Burr Brown, Tucson, AZ, USA. [Online]. Available: <https://www.digikey.se/en/products/detail/texas-instruments/INA114AU-1K/7033687>
- [12] Texas Instruments. (2019, Sep.) Tlv906xs 10-mhz, rrio, cmos operational amplifiers for cost-sensitive systems. Texas Instruments, Dallas, TX, USA. [Online]. Available: <https://www.digikey.se/sv/products/detail/texas-instruments/TLV9061IDBVR/9771994>
- [13] STMicroelectronics. (2022, Apr.) Stm32 32-bit arm cortex mcus. STMicroelectronics, Genève, Schweiz. [Online]. Available: <https://www.st.com/en/microcontrollers-microprocessors/stm32f091rc.html>
- [14] STMicroelectronics. (2020, Feb.) Description of stm32f0 hal and low-layer drivers. STMicroelectronics, Genève, Schweiz. [Rev. 7]. [Online]. Available: <https://www.st.com/en/embedded-software/stm32cubef0.html#documentation>
- [15] Curtiss-Wright Company. (2012, Dec.) Sls 095 linear displacement sensor. Curtiss-Wright Company, Davidson, NC, USA. [Online]. Available: https://www.cw-industrialgroup.com/getattachment/9812fa5a-5667-4ca2-887e-ae6521835305/sls095_technical_data

Design, Analysis and Implementation of a Drive System for Delsbo Electric Light Rail Vehicle

Maria Lindh and Daniel Marklund

Abstract—The aim of this project is to design and implement a drive system and a driving strategy for a lightweight, battery-driven rail vehicle partaking in the Delsbo Electric student competition. The goal of the competition is to create a vehicle which consumes as little energy as possible.

A simulation model of the vehicle is developed in Simulink, based on existing hybrid car models. Different drive cycles are written in MATLAB and tested in the vehicle simulation, which calculates energy consumption, power and torque usage and other important data. This data is used to select an optimal driving strategy and dimension the drive system components.

The final drive system design consists of a permanent-magnet synchronous motor powered by lead acid batteries and controlled by a microcontroller and motor driver through a user interface consisting of a control board with buttons and switches.

The chosen driving strategy combines slow acceleration and constant speed in slopes with the pulse and glide strategy on flat parts of the track. The simulation shows a total energy consumption of 0.67 Wh/person and km, which is in the same order of magnitude as results from previous years, which is promising for the competition. However, the actual energy consumption can not be known until the vehicle has been built and tested. There is a lot of uncertainty around its parameters at this stage, which affects the reliability of the simulations.

Sammanfattning—Syftet med det här projektet är att designa och implementera ett drivsystem och en körstrategi för ett lättviktigt, batteridrivet rälsfordon. Fordonet ska användas i studenttävlingen Delsbo Electric. Målet med tävlingen är att bygga ett fordon som förbrukar så lite energi som möjligt.

För att göra detta utvecklas en simuleringsmodell av fordonet i Simulink, baserat på redan existerande modeller av hybridbilar. Olika körprogram skrivs i MATLAB och testkör i modellen, som beräknar energiåtgång, använd effekt och vridmoment och annan viktig data. Dessa värden används sedan för att optimera körstrategin och dimensionera drivsystemets komponenter.

Det färdigdesignade drivsystemet består av en permanentmagnetiserad synkronmotor som matas från blyackumulatorer och styrs av en mikrokontroller och en driver via en kontrollpanel med knappar och switchar.

Den valda körstrategin kombinerar låg acceleration och konstant hastighet i backarna med pulse-and-glide-strategin på de platta delarna av banan. Enligt simuleringarna ger den en total energiåtgång på 0.67 Wh/person-km, vilket är i samma storleksordning som tävlingsresultat från tidigare år. Detta bådär gott inför tävlingen, men det går inte att veta hur stor den faktiska energiförbrukningen kommer bli förrän fordonet är byggt och testat. Än så länge är många av dess parametrar osäkra, vilket påverkar tillförlitligheten hos simuleringarna.

Index Terms—pulse and glide, rail vehicle, drive system, Delsbo Electric.

Supervisor: Mats Leksell

TRITA number: TRITA-EECS-EX-2022:138

I. INTRODUCTION

A. Background

Delsbo Electric is a student competition held annually in Delsbo, Sweden, by the non-profit organisation Dellenbanans Vänner. The purpose of the competition is to raise awareness about the efficiency of electric rail transportation and to inspire innovative solutions in the field [1]. Participating teams build rail vehicles for 1-6 passengers which have to be completely battery powered. The vehicles have to run on track between Fredriksfors and Delsbo (3.36 km) under 20 minutes. The winner is the one who uses the least amount of energy, measured in Wh/person and km [2].

This year, KTH has a team aiming to participate in the competition. The team was founded in 2020 and since then, important preparation work has been done. In 2020, a conceptual vehicle design was developed [3]. The next year, mechanical parameters were analysed dynamically and estimated using simulation software [4]. This year, electric power engineering students analysed and estimated electrical parameters and mechanical losses, and also compared two different driving strategies in terms of energy consumption [5]. The building of the actual vehicle has also started this year. The overarching goal for the whole KTH Delsbo team is to build a vehicle which can finish the competition on time and use no more than 1.5 Wh/person and km. This bachelor thesis project is set to aid in the development of its drive system and an efficient driving strategy for the competition, building on the research from previous years.

B. Problem formulation

The driving strategy is an important part of minimising energy consumption of a vehicle. According to [6], a common way to drive efficiently is to minimise the time spent accelerating and decelerating, and instead keep the speed as constant as possible. However, in some cases even less energy can be used by periodically accelerating above the desired speed and then turning off the engine. This technique, known as "pulse and glide" (PnG), has been thoroughly studied for combustion engine vehicles [7] but less research has been conducted on electric vehicles, although some research suggest that PnG can also reduce energy consumption in electric vehicles [8]. This project compares and combines both techniques to create a driving strategy optimised for this particular vehicle and running track. The following constraints on the driving cycle are set by the competition organisers [2]:

- The vehicle needs to reach the finish line 3.36 km from the starting point within 20 min.

- The average speed of the vehicle should be between 10 km/h and 15 km/h.
- The vehicle should come to a full stop within 20 m from the finishing line.

The drive system includes the motor and its power supply and control system. It is described more in detail in section II-A. The competition organisers have set the following constraints on the drive system:

- The battery must have a nominal voltage between 12 V and 48 V.
- The battery voltage should never exceed 60 V.
- Current consumption should never exceed 20 A.

The control system should combine manual and automatic control. Automatic driving should be the default but a manual override should always be an option in case the automatised driving fails. Furthermore the rules of Delsbo Electric require all vehicles to have an emergency handbrake.

This leads to the following project goals:

- 1) Develop a driving strategy optimised for the competition track.
- 2) Estimate drive system parameters, acquire components and install them in the vehicle.
- 3) Create and install a control system for driving the vehicle.

II. THEORY

A. Drive system

The drive system, illustrated in Figure 1, consists of a motor, battery, motor driver and a microprocessor. Gears, wheels and all other parts of the vehicle are outside the scope of this project and handled by the vehicle engineering team. The driver controls the vehicle with a control board connected to the microcontroller, which in turn is connected to a motor driver. The motor driver acts as a bridge between the motor and its power source, converting DC to AC and controlling the speed and torque of the motor by adjusting how much voltage and current it gets from the battery.

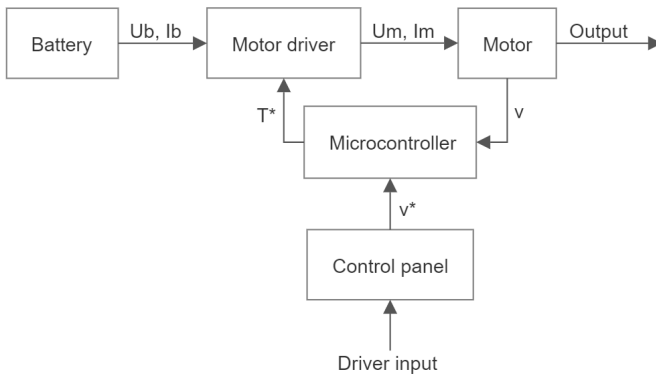


Fig. 1. Block diagram of the drive system. The microcontroller compares the motor speed v with the desired speed v^* to calculate how much torque is needed from the motor.

B. Energy efficient driving

By optimising the driving cycle it is possible to decrease the amount of energy consumed during the competition. There are many factors that influence the end result. If the operation power is the only power consumption considered, the logical conclusion would be to finish the race as fast as possible to minimize the run time and thereby minimizing the power consumed by the electronics. However, a higher velocity will also lead to increased drag which means that driving slower would save more energy if that is the only factor considered. A common method found when reading about efficient driving is pulse and glide. Pulse and glide is a driving method where the vehicle is first accelerated to a certain speed. Then it decelerates until a threshold is reached and restarts the process. This method has been proven to save energy in systems using petrol as well as electricity according to [9] and is therefore of interest when determining the optimal drive cycle from an energy conserving perspective. When optimising the pulse and glide drive cycle it is important to consider how fast the system accelerates and decelerates as well as the highest and lowest speed of the vehicle. To be able to reclaim some of the energy spent while driving electrical vehicles can make use of something called regenerative braking. Regenerative braking is a method that is sometimes used in systems with electric motors to conserve energy. When braking regeneratively the motor is essentially being used as a generator. The idea is to make use of the kinetic energy and any force contributed from a slope to make the axis of the motor spin. The spin will generate a voltage increase at the output of the battery, making it charge [10]. It is especially interesting when going downhill since it is then possible to convert potential energy into electricity without any acceleration from the vehicle.

C. Synchronous machine

Synchronous machines consist of a stator and a rotor, as shown in Figure 2. The rotor is either a permanent magnet or an electromagnet and it upholds a magnetic field in the motor. The stator contains windings powered by an AC supply, giving rise to another magnetic field. This field rotates with the same frequency as the AC supply, which causes the rotor to rotate as well as it aligns its magnetic field with the stator field. The name 'synchronous' comes from the fact that the rotor movement is synchronous with the electrical supply frequency. The synchronous machine can also be used as a generator: an external torque exerted on the rotor will cause its magnetic field to rotate, inducing an electromotive force and current in the stator coils [11].

III. METHOD

In the beginning of the project a simulation model of the vehicle was developed. This was used to test different drive cycles and calculate the requirements on the drive system. When all components had been ordered the control system was designed.

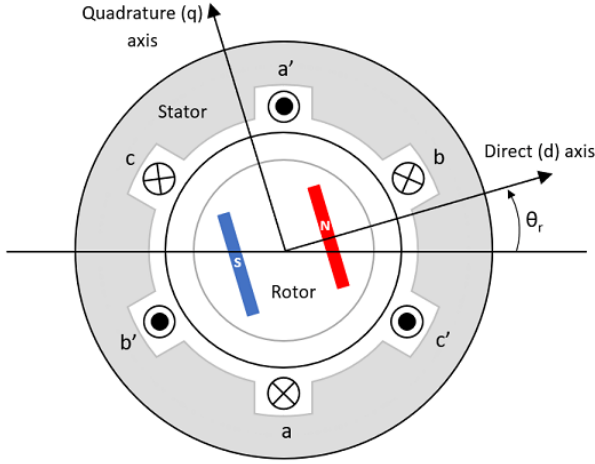


Fig. 2. Three phase permanent magnet synchronous motor (PMSM). Source: [12].

A. Vehicle simulation model

The vehicle was modelled in Simulink. The model was developed from a series hybrid car model from the course Hybrid Vehicle Drives EJ2410 at KTH. It was remade into a fully electric vehicle with appropriate parameters. It could follow a given driving cycle and provide information about energy consumption and usage of power and torque over the whole cycle. The model was used to test different driving cycles and find demands on the motor in terms of power, torque and speed. A simplified block diagram of the model is shown in Figure 3 and the full code is available in Appendix A.

The driver was modelled as a PI controller providing the desired force

$$F^* = K_v \left((v^* - v) + \frac{1}{T_i} \int_0^t (v^* - v) dt \right) \quad (1)$$

from the desired speed v^* according to the driving cycle, and the actual speed v . The controller parameters were dependent on the vehicle mass m in the following way:

$$T_i = 3 + 7 \frac{m - 1000}{9000}, K_v = 4 \frac{m}{T_i} \quad (2)$$

This was established in the original model from the course EJ2410 and kept the same in this project. In the transmission block, F^* was transformed into a demanded wheel torque

$$T_{wheel}^* = F^* \cdot r_{wheel} \quad (3)$$

where $r_{wheel} = 0.15$ m was the wheel radius of the vehicle. This was subsequently used to determine the motor torque demand

$$T_{motor}^* = \frac{T_{wheel}^*}{gr \cdot \eta_{gear}} \quad (4)$$

where $gr = 9 : 1$ was the gear ratio and $\eta_{gear} = 75\%$ the gear efficiency. Motor angular velocity

$$\omega_{motor} = gr \frac{v}{r_{wheel}} \quad (5)$$

was calculated from the vehicle speed v and provided motor torque

$$T_{motor} = \min(T_{motor}^*, T_{motor,max}) \quad (6)$$

was limited by the motor's max output torque which was set when initialising the model. The total provided power

$$P_{motor} = \frac{T_{motor} \cdot \omega_{motor}}{\eta_{motor} \cdot \eta_{PE} \cdot \eta_{gear}} \quad (7)$$

was also determined in the motor block. The brake compared demanded and provided torque when slowing down and provided the negative torque needed whenever $T_{motor} > T_{motor}^*$ without limitations. This was not realistic, but since the vehicle was only running at low speeds and the goal was to not use the brake at all, it did not matter for the simulations.

In the track model block, the actual vehicle speed v was determined by integrating the acceleration coming from the sum of all forces acting on the vehicle. Some of these forces, which are explained more in detail in section III-B, were dependent on v itself.

The battery block was not "powering" anything in itself. Its energy content was set when initialising the model and during simulation its drainage was calculated by integrating P_{motor} . When it reached a set minimum value the simulation stopped.

B. Modelling losses

The vehicle model accounted for seven different types of losses:

- 1) Rolling resistance $F_{rolling}$
- 2) Air resistance F_{air}
- 3) Slope resistance F_{slope}
- 4) Microcontroller power supply P_{mc}
- 5) Power electronics efficiency η_{PE}
- 6) Electrical motor efficiency η_{motor}
- 7) Gear efficiency η_{gear}

Values for air and rolling resistance were approximated in simulations done by the vehicle engineering team. The rolling resistance was estimated to 0.2 N and 2.4 N on straight and curved parts of the track respectively. By using the approximation shown in Figure 4, the average rolling resistance was estimated as a weighted average to

$$F_{rolling} = 2.4 \frac{500}{3360} + 0.2 \frac{2860}{3360} \approx 0.53 \text{ N} \quad (8)$$

The air resistance, which depended on the vehicle shape and speed, was estimated to

$$F_{air}(v) = 0.256 \cdot v^2 \quad (9)$$

where v was the speed of the vehicle. Slope resistance was given by

$$F_{slope}(x) = mg \sin(\alpha(x)) \quad (10)$$

where x was the distance travelled from the starting point in Fredriksfors, $m = 420$ kg was the total weight of the vehicle, $g = 9.82$ m/s² was the gravity, and α was the slope at a particular point of the track.

The microcontroller (Raspberry Pi 4B) had an idle power consumption around 2 W, which could increase to up to about 5 W under load [14]. It was set to a constant $P_{mc} = 5$ W.

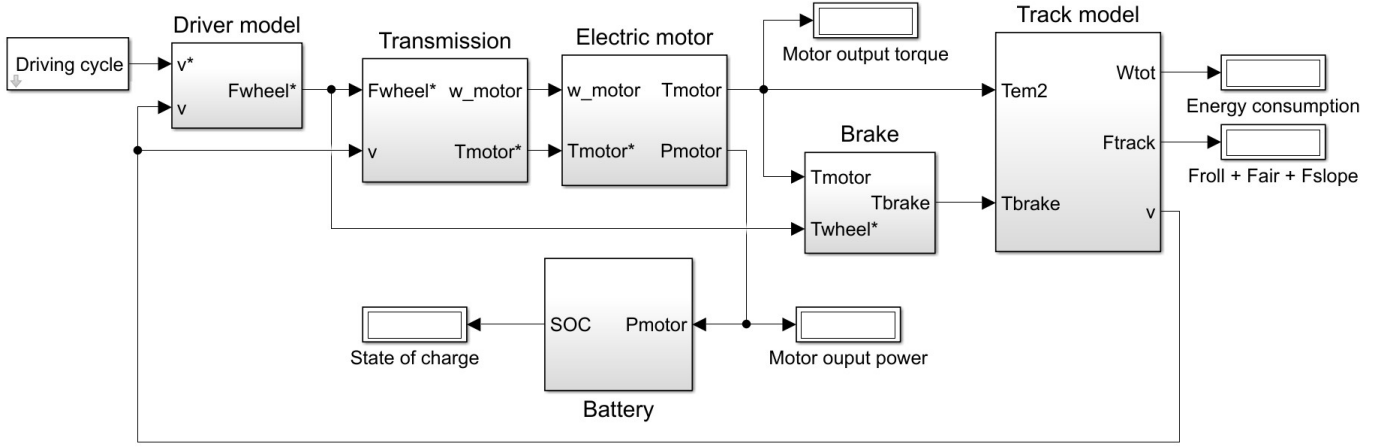


Fig. 3. Block diagram of the Simulink vehicle model.

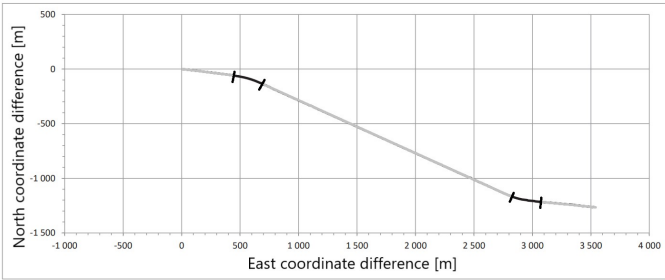


Fig. 4. The competition track from above with Delsbo station at the origin. Grey parts are approximated as straight lines and the black parts are the only curves of the track. The curved parts make up around 500 m of the 3360 m long track. Source: [13]

The motor driver contained an inverter which transformed DC power from the battery to AC power for the motor. Such converters typically have a very high efficiency [10], but in the vehicle model it was conservatively estimated to $\eta_{PE} = 80\%$.

Electrical motors have many different types of losses. They can be divided into four groups: copper (ohmic), iron (magnetic), mechanical and stray losses. Copper loss or ohmic loss occurs whenever a current I passes through the motor wires with resistance R , according to

$$P_c = RI^2 \quad (11)$$

Since motor torque T_{motor} is proportional to the current, P_c will be proportional to T_{motor}^2 .

Iron losses or magnetic losses are losses that occur in the iron core of the rotor due to the constantly changing magnetic field. Some energy is lost when magnetic dipoles in the iron realign themselves with the moving field. This is called hysteresis loss and is given by

$$P_h = \eta \hat{B}^n fV \quad (12)$$

where f is the frequency of the magnetic field, V is the core volume and \hat{B} is the peak flux density. Steinmetz hysteresis coefficient η and Steinmetz exponent $n \in [1.5, 2.5]$ both depend on the material. The other type of iron loss is called

eddy current loss, from the eddy currents that arise in the magnetised iron in accordance with Faraday's law. These currents produce ohmic losses in the iron given by

$$P_e = k_e \hat{B}^2 f^2 d^2 \quad (13)$$

where d is the thickness of the iron in m and k_e is a constant that depends on volume and resistivity. Both P_h and P_e depend on f , which is proportional to the speed of the motor.

Mechanical losses refer to friction and windage losses. Friction will arise in all moving parts of the motor that touch each other, primarily the bearings. Windage loss refers to wind resistance inside the motor coming from the moving rotor. Both types depend on the motor speed.

The remaining loss types are collectively referred to as stray losses. These have several origins but are usually estimated to only make up a small part of the total power loss [10] [11].

Based on these losses, the vehicle model calculated the motor's efficiency as a function of its speed and torque, which is shown in Figure 5.

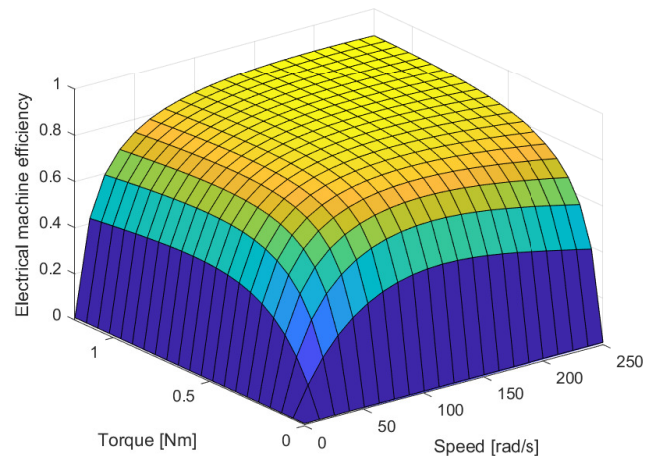


Fig. 5. Electrical motor efficiency as a function of speed and torque.

For the gearing a belt gear with a gear ratio of 9 : 1 was used. This was done by the vehicle engineering team, who estimated its efficiency to $\eta_{gear} = 75\%$.

C. Running track height profile

Altitude measurements of the track were provided by the competition organisers [2]. This data was linearised into seven parts with constant slope, as shown in Figure 6. This linearisation was used to calculate the slope force used in the vehicle simulation, and to divide the track into distinct parts for optimisation of the driving cycle.

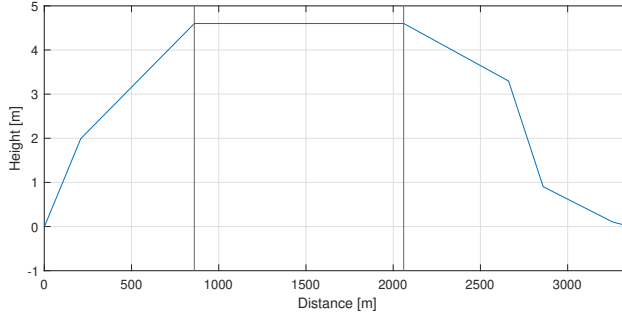


Fig. 6. The linearised altitude profile of the competition track. The vertical lines show division into uphill, plane and downhill.

D. Drive cycle optimisation

A driving cycle shows speed as a function of time for a predefined time interval. In this project, a drive cycle of length t seconds was represented by a $t \times 2$ matrix, where the first column contained the time vector from 1 to t seconds and the second column showed speed in m/s for each second.

As mentioned previously, the track was divided into seven parts with approximately constant slope. The drive cycle was to be optimised on each part separately and combined into a complete cycle for the whole track. However, in a first phase of the optimisation process these parts were combined into only three groups: uphill, plane and downhill slope, as shown in Figure 6. The speeds used were also restricted to $v_{slow} = 10$ km/h, $v_{medium} = 12.5$ km/h and $v_{fast} = 15$ km/h. This was done to simplify the process. On each part, one of the following driving strategies was implemented: constant acceleration followed by a constant speed, or constant acceleration followed by PnG around a constant speed. For each driving strategy there were a number of parameters that had to be determined.

Constant speed:

- constant acceleration a_0
- constant speed v_0

PnG:

- constant acceleration a_0
- average speed v_0
- pulse/glide amplitude v_{amp}
- pulse acceleration a_{pulse}
- glide acceleration a_{glide}

Different driving cycles were made and tested by running them in the Simulink model. For each cycle the total energy consumption was noted, but also peak power and torque used during the run. These values were used when determining requirements on the motor.

In the first optimisation phase, one parameter at a time was studied. When a 10% change of a parameter did not impact the energy consumption more than 10% it was considered optimised and brought to the next stage where a different parameter was studied. On each part of the track (uphill, plane, downhill), parameters were set in the following order:

- 1) v_0 on each part, with a set acceleration of $a_0 = 0.25$ m/s².
- 2) $|a_0|$ (same on every part).
- 3) Constant speed or PnG with $v_{amp} = 0.5$ km/h, $a_{pulse} = 0.05$ m/s², $a_{glide} = 0.01$ m/s² for every part.
- 4) PnG parameters (v_{amp} , a_{pulse} , a_{glide}) for the parts where PnG was used.

When the first phase was done, all parameters were surprisingly stable. For example, a 40% change in the downhill speed from 10 km/h to 14 km/h did not increase the energy consumption more than 6.9%. It was concluded that continuing to the second phase and make the optimisation even more detailed would not have a significant impact on the energy consumption. Other factors such as vehicle weight, wheel type and gear belt friction would have a much larger impact. Therefore, the second optimisation phase was omitted to save time for other parts of the project.

E. Determining component parameters

The first component to be acquired was the microcontroller. During the project there was a worldwide shortage of electric components so rather than waiting for an optimal one, the first one available was bought. This was a Raspberry Pi 4B, and typical values for its power consumption were used in the simulation model of the vehicle. Near the end of the project it was changed into an Arduino Uno with lower power consumption.

Motor and battery parameters were determined in parallel with the development of the driving cycle. In the Simulink vehicle model, graphs of torque and power from the motor were plotted for each driving cycle. These graphs were used to determine nominal torque and nominal output power needed from the motor. These were the most important motor parameters, and some driving cycles had to be discarded despite superior energy consumption because their torque and/or power requirements were too high. In addition to these constraints, the motor also had to have input voltage and current below or equal to 48 V and 20 A respectively, to fulfill the criteria set by Delsbo Electric (see section I-B). Output speed was not an issue since the vehicle was not meant to go above 15 km/h, which was definitely achievable by most motors on the market. Apart from this, the goal was to have a motor with as low weight as possible, since weight has a large impact on the energy consumption of a vehicle.

The constraints on the battery were that it needed to output enough voltage to drive the motor, while not going above the

competition criteria of $V_{max} = 60 \text{ V}$. It also had to have a capacity greater than the total energy consumption during the competition run. To be on the safe side, a margin of ten times the simulated energy consumption was set.

The next step was to determine driver parameters, since they depended on both the battery and the motor. It also had to be compatible with the microcontroller. Fortunately, the company supplying the motor also provided a suitable driver so this was not a problem.

F. Control algorithm

To be able to control the vehicle the driver was given access to a handful of pre-programmed driving cycles that could be switched between with the help of buttons or switches. An override had to be implemented into the system due to regulations and uncertainty about the track. The rules of the competition demanded that there would be a way to stop the system by hand using a brake without any risk of injury for the people riding the vehicle. To ensure safety and lessen the likelihood of damage to the system, an override was implemented into the system. This override activates whenever the brake is used and forces the system to abort the current driving instructions and turn off the motor. The override was implemented by designing the program with hierarchy in mind. The program was designed according to figure 7 with the emergency break at the top of the hierarchy. One step down, the manual control is found. This function was implemented to give the driver the ability to compensate for inaccuracies in the simulation by accelerating and braking. Further down the different driving cycles can be found. The driver was able to enter these modes by flipping a switch. The driving cycles themselves were stored as speed values in arrays. A PI-regulator was implemented in the code to make the motor follow the driving cycles. The PI regulator compared the speed of the vehicle with the reference speed in the current driving cycle and calculated a signal for desired motor torque which was sent to the driver. This is shown in Figure 1. In the case where no switch was set to ON the system was told to stop just as when the emergency brake was pulled. The program was written in C++ for the Arduino and the code is available in Appendix B.

IV. RESULTS

In this section, the final drive cycle and components are presented, and the plan for the control system is described.

A. Driving cycle

The final driving cycle is shown in Figure 8. Uphill and downhill the vehicle accelerates or decelerates with 0.04 m/s^2 to a constant speed of 10 km/h uphill and 11 km/h downhill. On the flat part of the track PnG is used with the parameters of Table I. It reaches the finish line in 1125 s and has a total energy consumption of $0.67 \text{ Wh/person and km}$.

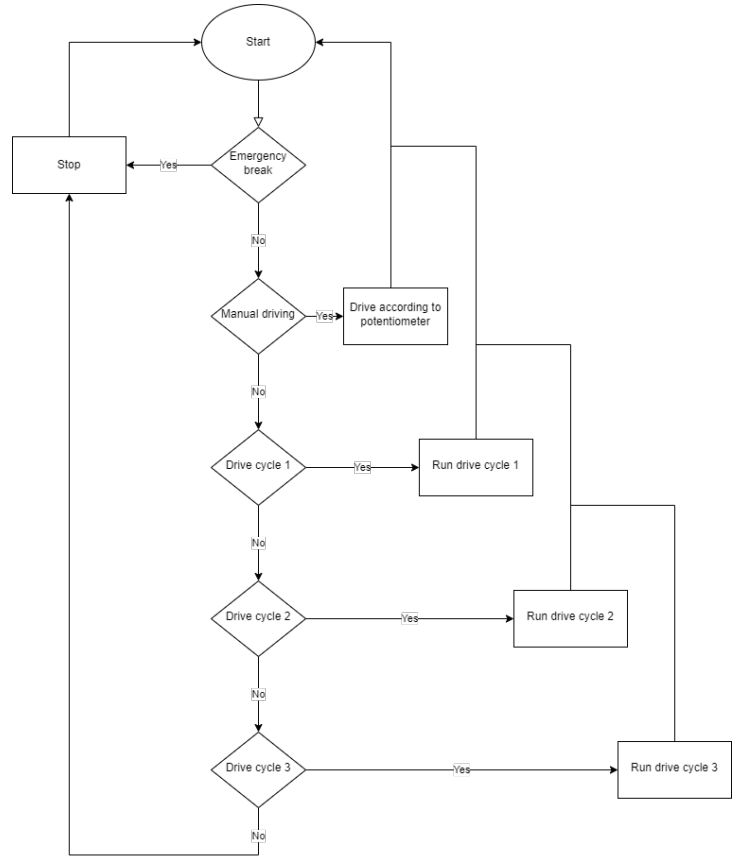


Fig. 7. Flowchart showing the general structure of the program.

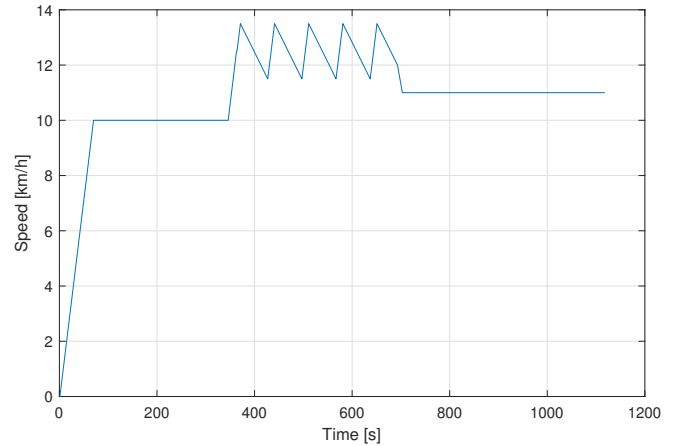


Fig. 8. The final drive cycle.

B. Components

1) *Microcontroller*: The chosen microcontroller is an Arduino Uno. It is robust and easy to use, but has less functionality than the Raspberry Pi. It also has a lower power consumption.

2) *Motor*: The motor used is a HDD 09E servomotor, which means that it is very precise. More specifically, it is a three phase permanent magnet synchronous motor with specifications given in Table II. Further information can be

TABLE I
PNG PARAMETERS

Parameter	Value
average speed v_0	12.5 km/h
pulse/glide amplitude v_{amp}	1 km/h
pulse acceleration a_{pulse}	0.04 m/s ²
glide acceleration a_{glide}	-0.01 m/s ²

found in the datasheet [15].

TABLE II
MOTOR PARAMETERS

Parameter	Value
Rated torque	1.2 Nm
Rated output power	377 W
Input voltage	48 V
Rated current	1.1 A
Weight	1.8 kg

3) *Battery*: The motor needs 48 V and the total energy consumption for the chosen driving cycle is 13.6 Wh. The cheapest solution is to use four 12 V, 7.2 Ah lead acid batteries from Biltema [13]. Together they have a total output voltage of 48 V and a capacity of 345.6 Wh.

4) *Motor driver*: The motor driver is customized according for the motor and battery and was provided to the project by HDD Sweden AB. The motor driver provided is the XtrapulsEasy 60 VDC DB. This driver supports voltages up to 60 V and can be controlled using PC-software, CAN bus or analog signal input [16].

C. Control system

The program is written in C++ and implemented on the previously mentioned microcontroller. The program first checks the output from the input device and based on the input runs the appropriate instructions. Depending on the selected instruction the program will check the feedback from the motor driver and adapt to ensure that the desired functionality is achieved.

V. DISCUSSION

A. Vehicle modelling

Some of the model parameters were quite uncertain, particularly rolling and air resistance, gear and power electronics efficiency, and microcontroller power consumption. The last three could have been tested to improve the results, but rolling and air resistance can not be tested until the vehicle is built and running. Hopefully this can be improved in future projects.

B. Motor selection

In the end the motor that was selected was not selected due to its low energy consumption but because of its low cost. Both the motor and the driver were lent to the project by HDD motors which meant that a lot of budget could be freed up for the other groups, which was of interest due to the low budget for the project. It was decided that it would be more

valuable to redistribute the resources to other groups than to spend money on a motor that would be more fitting for the task.

C. Microcontroller power consumption

During this project the main focus was on minimizing the power consumption from the drive system, while losses in the microcontroller were of lower priority. The main reason for this was a lack of available components. To ensure that the system would be working before the deadline of the competition, a Raspberry pi 4B was bought since it was the only one on the market at the time. This option was very powerful but it consumed quite a lot of energy compared to other microcontrollers. In tests the Arduino Uno has been shown to consume as little as 150mW [17] and the Raspberry Pi 3B has been shown to consume as little as 1.4 W. Both of these results were considerably lower than the minimum 2.7 W demanded by the Raspberry pi 4B. A Raspberry Pi 3B+ was lent to the project later on slightly lowering the power consumption, and towards the end of the project it was changed once more to an Arduino Uno that one of the project group members had available from previous projects. In the future, more research could be made on which microcontroller would be optimal for the vehicle. At this point the control system is not very complex, which means that a simple microcontroller can be used. Perhaps the Arduino Uno can be replaced by an even more energy efficient controller in the future. The power consumption could also be lowered by optimising the code to reduce processing power demands on the microprocessor.

D. Drive cycle optimisation

The initial plan in this project was to optimise the driving cycle for each of the seven parts of the track with distinct slope. After running the first optimisation phase with just three track divisions, it was clear that small changes in the speed barely had a noticeable impact on the energy consumption. Therefore, it was decided to skip the second phase, to save time and focus on other parts of the project. However, one could argue that even small percentages matter and there was clearly still room for improvement. Another thing that could be improved was the optimisation process itself. Instead of manually creating and testing new driving cycles, the process could have been automatised and some optimisation algorithm could have been used. For example, the genetic algorithm was discussed in the beginning of the project. However, none of this was implemented since manual testing produced good results quickly. The external target of 1.5 Wh/person and km was surpassed with good margin early on so it was decided to prioritise spending time on other parts of the project instead of optimising further. However, this is something that could be investigated in future projects.

Another thing worth noting is that the chosen driving cycle was not the one with the lowest energy consumption. There were others who were even lower, but they required much higher power and torque from the motor, which was not possible to achieve within the budget.

E. Automation of control

The vehicle was controlled using a few different predetermined modes and manual control. However the ambition in the future is to have a fully automated vehicle. This would be beneficial since a computer could process information from the sensors directly and make more optimal decisions than any driver could consistently produce. To achieve this goal a few features would have to be added. The system has to be able to read its position geographically and also read the conditions of the track. By knowing the system's geographical position it would be possible for the computer to follow a strategy and not only drive according to the current conditions. But knowing the current conditions is important since it enables the system to control itself in an efficient manner. Once these features have been added to the system it would be possible for the system to control itself by following the same algorithm as before. This automation would also make it possible to use more advanced instructions that could be difficult for a human to follow, potentially lowering the energy cost.

F. Competition

There were some limitations regarding which factors were considered in the competition. The measurement of energy consumption only concerned itself with power coming out and going into the battery but does not consider how this affected the battery or if the battery was actually getting charged to begin with. It would be in the spirit of the competition to consider these effects and work should be done in the future in anticipation of this since it may very well be controlled in future competitions.

VI. CONCLUSION

The final drive system consists of a HDD 09E servomotor [15], an XtrapulsEasy 60 VDC DB driver [16], four lead acid batteries [13] and an Arduino Uno [18]. Its driver interface consists of a control panel with options for both manual and automatic driving, where the automatic option has three different drive programs for the different parts of the competition track: positive slope, negative slope and flat track.

The ideal driving cycle is shown in Figure 8 and according to the simulations it has a total energy consumption of 0.67 Wh/person and km. It was shown that PnG is beneficial to use on the flat part, but that constant speed is better both uphill and downhill in terms of minimising energy consumption. It was also shown that decreasing speed typically decreased energy consumption as well, but only down to a certain limit due to decreased motor efficiency.

APPENDIX A

SIMULINK VEHICLE MODEL

APPENDIX B

CODE FOR THE CONTROL SYSTEM

ACKNOWLEDGMENT

First and foremost the authors would like to thank their supervisor Mats Leksell for his continuous support throughout the project. His guidance has been of great value when

overcoming all of the hurdles that arise in practical application. The authors would also like to thank the rest of the KTH Delsbo team for their diligent work and for being a welcoming group for new members. The authors would like to thank Patrick Janus for the support during the construction as well as connecting us to one of our main sponsors. Lastly the authors would like to thank their sponsors STHK and HDD for providing the resources necessary to turn this project into a reality and enabling KTH to compete in Delsbo electric for the first time.

REFERENCES

- [1] (2022, Mar) Delsbo electric. [Online]. Available: <https://www.delsboelectric.se/omoss>
- [2] (2022, Mar) Regler 2022. [Online]. Available: <https://www.delsboelectric.se/regler2022>
- [3] P. Geiberger, G. Holmström Praesto, and S. Ram Aravindababu, "Delsbo electric: Concept design for a rail vehicle with high energy efficiency," KTH Royal Institute of Technology, Tech. Rep., Oct 2020.
- [4] G. Holmström Praesto, "Delsbo electric: Dynamic analysis of a rail vehicle with high energy efficiency," KTH Royal Institute of Technology, Tech. Rep., Jan 2021.
- [5] C. Henrikson, P. Subramaniyane, R. Jain, R. Raj, and S. Schneider, "Design of an energy efficient electric drive system for a railway contest," KTH Royal Institute of Technology, Tech. Rep., 2022.
- [6] A. Sciarretta and A. Vahidi, *Energy-Efficient Driving of Road Vehicles: Toward Cooperative, Connected, and Automated Mobility*. Cham, CH: Springer International Publishing AG, 2019.
- [7] J. Kim and C. Ahn, "Real-time speed trajectory planning for minimum fuel consumption of a ground vehicle," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 6, pp. 2324–2338, Jun 2020.
- [8] K. M. So, P. Gruber, D. Tavernini, A. E. H. Karci, A. Sorniotti, and T. Motaln, "On the optimal speed profile for electric vehicles," *IEEE Access*, vol. 8, pp. 78 504–78 518, Mar 2020.
- [9] Z. Tian, L. Liu, and W. Shi, "A pulse-and-glide-driven adaptive cruise control system for electric vehicle," *International transactions on electrical energy systems*, vol. 31, no. 11, 2021.
- [10] J. Larminie and J. Lowry, *Electric vehicle technology explained, second edition*, 2nd ed. Chichester, West Sussex, U.K: Wiley, 2012.
- [11] H.-P. Nee, M. Leksell, S. Östlund, and L. Söder, *Eleffektsystem: EJ1200*. Stockholm, Sweden: KTH Royal Institute of Technology, School of Electrical Engineering and Computer Science, 2019.
- [12] MathWorks. (2022, Apr) Interior pmsm. [Online]. Available: <https://jp.mathworks.com/help/mcb/ref/interiorpmsm.html>
- [13] *Blyackumulator*, Biltema, Apr 2021. [Online]. Available: <https://www.biltema.se/bil---mc/bilbatterier/blyackumulatorer/blyackumulator-12-v-72-ah-151-x-65-x-95-mm-2000047907>
- [14] (2022, May) Power consumption benchmarks. [Online]. Available: <https://www.pidramble.com/wiki/benchmarks/power-consumption>
- [15] *HDD 09E*, HDD, Jul 2014. [Online]. Available: <https://hdd.se/servo-motors/solid-shaft-servo-motors/solid-shaft-motors-90-frame/hdd-09e/>
- [16] *XtrapulsEasy 60 VDC DB*, Infranor, May 2021. [Online]. Available: https://infranor.com/en/downloads/#elf_11_UHJvZHVjdHMvU2VydM8tZHJpdmVzL1h0cmFwdWxzRWFZeS9NYW51YWxz
- [17] M. Al-Shorman, M. Al-Kofahi, and O. Al-Kofahi, "A practical microwatt-meter for electrical energy measurement in programmable devices," *Measurement and Control*, vol. 51, p. 002029401879435, 08 2018.
- [18] *Arduino UNO R3*, Arduino, May 2022. [Online]. Available: <https://docs.arduino.cc/hardware/uno-rev3>

Minnestekniker bortom halvledare för inbyggda system

Taseen Chowdhury

Abstract—Silicon manufacturers are experiencing shortages of semiconductors and the demand for cost-effective, power-efficient embedded memory solutions is increasing. For these issues, a new emerging memory technology called embedded magnetoresistive random access memory (eMRAM) and the development of the write mechanism called spin-transfer torque (STT-MRAM) have been proposed. The eMRAM has non-volatility, reduced total energy consumption, fast read/write operation and has a small macro size compared to the semiconductor-based memory types such as SRAM, Flash and EEPROM. The purpose of this study is to investigate eMRAM and how it can be used in a microcontroller to replace all three existing, semiconductor-based memory types. The focus will be on how solution can be created with smaller memory chip area, improved energy efficiency and faster read/write operations. A literature review was established, to determine if eMRAM does indeed result in better memory characteristics and memory performance. As well as to determine the requirements that is needed for a flash-type and SRAM-type application. The study shows that eMRAM have a potential to create many solutions for a microcontroller, such as it has the potential to simplify its memory architecture by providing a unified memory solution for its code and data storage as well as for its working memory.

Sammanfattning—Halvledartillverkarna står inför svårigheter på grund av bristen på halvledare och att efterfrågan på kostnadseffektiva, strömsnåla inbyggda minneslösningar ökas. För att lösa dessa problem har en ny framväxande minnesteknik som kallas för inbyggd magnetoresistivt slumpmässigt åtkomstminne (eMRAM) och utvecklingen av skrivmekanismen som kallas för spinn-överföringsmoment (STT-MRAM) föreslagits. eMRAM har icke-flyktighet, en låg total energiförbrukning, snabba läs- och skrivfunktioner och har en liten makrostorlek jämfört med de halvledarbaserade minnestyperna såsom SRAM, Flash och EEPROM. Syftet med denna studie är att undersöka eMRAM och hur det kan användas i en mikrokontroller för att ersätta alla tre befintliga halvledarbaserade minnestyper. Fokuset kommer att ligga på hur en lösning kan skapas med mindre minneschipyta, bättre energieffektivitet och snabbare läs- och skrivoperationer. En litteraturgenomgång gjordes för att fastställa om eMRAM verkligen resulterar i bättre minnesegenskaper samt minnesprestanda och att fastställa de krav som krävs för en tillämpning av en flash-typ och en SRAM-typ applikation. Undersökningen visar att eMRAM har en potential att skapa många lösningar för en mikrokontroller, t.ex. har den som potential att förenkla dess minnesarkitektur genom att bidra med en enhetlig minneslösning för kod- och datalagring samt för arbetsminnet.

Nyckelord—MRAM, eMRAM, STT-MRAM, mikrokontroller, MTJ, spintronik.

Handledare: Gunnar Malm

TRITA nummer: TRITA-EECS-EX-2022:139

I. INTRODUKTION

Åren 2020-2022 var världen i en pågående global pandemi och som en konsekvens behövde flera företag i världen, utifrån ett hälsoperspektiv anpassa deras verksamheter. Pandemin orsakade därav en mindre efterfrågan av komponenter i världen och för den redan drabbade halvledarindustrin innebar det en global brist på halvledarkomponenter, vilket även orsakades av den låga marknadsandelen. Då en halvledartillverkare som mest har en total försäljning som uppnår 15% i halvledarindustrin. Det beror på att industrins verksamheter är beroende av varandra och specifika komponenter som tillverkas för inbyggda system är uppdelade bland de få halvledartillverkarna [1].

Inbyggda system har alltid varit en vital del av vår digitala infrastruktur. Enligt [2] är inbyggda system ett mikrodator baserat system, som har en viktig funktion för ett större mekaniskt eller elektriskt systems funktionalitet. De två vanligaste mikrodatorerna för inbyggda system är en mikroprocessor och en mikrokontroller, vars syfte är att utföra specifika funktioner och som har bidragit en mångfald av funktionalitet, intelligens och säkerhet inom fordonsindustrin och för modern teknologi. Enligt [1] används upp till 98% av alla tillverkade mikrokontroller och mikroprocessor i produkter såsom fordon, hushållsapparater och i nästan alla elektroniska system. Det har dessutom diskuterats om inbyggda systems betydelse för utvecklingen av artificiell intelligens (AI) och maskininlärning (ML), som har drivit på att dessa trender har blivit möjligt för den moderna teknologin. Det finns idag flera olika mikrokontroller som har konstruerats och specificerats utifrån behovet av det större inbyggda systemets specifikationskrav och systemkrav. Men då variationen och komplexiteten för elektroniska system är i ett ständigt växande fas och att tillverka specifika mikrokontroller inte är möjligt längre, har detta utvecklat ett behov för avancerade mikrokontroller som möjliggör en effektiviserad energiförbrukning och en förbättrad systemprestanda för ett inbyggt system [3].

A. Problembeskrivning

Inbyggda system såsom mikrokontroller består generellt av en minnesarkitektur, som har ett programminne och ett slumpmässigt åtkomst dataminne [4]. Minnesarkitekturen består därför vanligtvis av tre inbyggda minnestyper, statistiskt slumpmässigt åtkomstminne (SRAM), FLASH och elektrisk raderbar programmerbar läsminne (EEPROM) [5]. I flash-minnet lagras kodinstruktioner och programdata som inte ändras efter uppstart och i EEPROM lagras data av variabler. De båda minnestyperna är icke-flyktiga minnen, som innebär att programminnets lagring av data inte försvinner

när systemet saknar matningsspänning och kan därför bevara informationen i mer än 10 år. SRAM används för att utföra samt lagra instruktioner och data under drift. Minnestypen utför snabba minnesoperationer som läs- och skrivoperationer i nanosekunder och har obegränsad antal programmeringscykler samt är ett flyktigt minne. Det innebär att minnestypen inte kan behålla det lagrade data när systemet saknar matningsspänning [6]. Dock är dessa inbyggda minnen halvledarbaserade och lagrar därför data genom en elektrisk laddning i minnestypernas minnesceller, som består av metalloxid-halvledare fälteffekttransistorer (MOSFET) [7]. Dessutom har de olika typer av minnesegenskapsbegränsningar såsom att en SRAM minnescell består av flera transistorer, vilket begränsar minnets skalbarhet och det innebär att SRAM använder mycket chiparea i en mikrokontroller minnesarkitektur. Dessutom använder SRAM mycket energi för att utföra snabba minnesoperationer [8] [9]. Flashminnet och EEPROM har begränsad antal programmeringscykler och är långsamma, för flashminnet framgår det att minnet utför blockvis skrivningar som bidrar för en långsammare minnesoperationstid [10].

En alternativ minnestyp som har utvecklats och som inte är direkt baserad på halvledare är inbyggd magnetoresistivt slumpmässigt åtkomstminne (eMRAM). eMRAM är en spintronik baserad minnestyp vars minnesteknik använder elektronernas spinn som genererar ett magnetiskt energimoment i två riktningar, spinn-upp av elektroner och spinn-ner av elektroner. Det gör att eMRAM kan lagra data i ett magnetiskt lagringselement med konfigurationstekniken, magnetisk tunnelkorsning (MTJ). eMRAM har liknande minnesegenskaper som både ett flyktigt minnestyp samt ett icke-flyktigt minnestyp på grund av att elektronernas spinn inte behöver en konstant spänning. För eMRAM används den utvecklade minnestekniken, spinn-överföringsmoment (STT) MRAM minnet. STT är en ströminducerad skrivmekanism som möjliggör snabba minnesoperationer, en hög databevaringstid och en obegränsad minnesuthållighet för minnestypen. Minnesegenskaperna för eMRAM medför att minnet har en mindre chiparea, bättre energieffektivitet och snabba minnesoperationer för en mikrokontroller.

Trots att STT-MRAM är en lovande minnestyp för ett minskat halvledarberoende och samtidigt vara en lösning för en mikrokontroller minnesbegränsningar, medför minnestypens minnesegenskaper att framför allt energiförbrukningen för en skrivoperation har behövt kompromissats för att uppnå icke-flyktiga egenskaper. Det betyder att eMRAM kan bidra med en högre energikostnad än de tidigare halvledarbaserade minnestyperna i en mikrokontroller, vilket är ett problem [9] [3]. Ett ytterligare problemområde är att STT-MRAM är en ny utvecklad minnesteknik och det betyder begränsade kunskaper som kommersiell minnestyp. Dessutom har en mikrokontroller med en minnesarkitektur baserad på minnestypen ännu inte nått den kommersiella marknaden, utan att den endast är i massstillverkningsstadiet, som en ensamstående minneschip [11]. Det är en fråga om tillförlitlighet, då de tidigare halvledarbaserade minnestyperna inte har ett sådant problem oavsett deras begränsningar, utan att det istället är en fråga om utvecklingsbegränsningar [8] [12]. Det betyder att en fördjupning angående eMRAMs pålitlighet, minnesprestanda

samt den totala energiförbrukningen måste undersökas och vilka lösningar som skapas, som en ersättande minnestyp för en mikrokontroller minnesarkitektur.

B. Mål

Målet med projektet är att göra en grundlig undersökning av eMRAM genom att ge en detaljerad beskrivning på eMRAM funktionalitet, egenskaper, begränsningar och minnestekniken bakom minnestypen. Dessutom är projektets mål att undersöka hur det kan användas i en mikrokontroller för att ersätta alla tre befintliga, halvledarbaserade minnestyper. Projektet syftar till att besvara frågeställningarna på hur man kan skapa lösningar med mindre chiparea, bättre energieffektivitet och snabbare läs- och skrivoperationstider med eMRAM för inbyggda system såsom mikrokontroller.

C. Metod

Projektet är upplagt som en litteraturgenomgång till stor del från öppna källor. För att besvara frågeställningarna i projektet behövs en grundlig undersökning genom en litteraturstudie görs för minnestypen eMRAM. För en sådan studie behövs trovärdiga källor såsom vetenskapliga artiklar, litteraturer och konferensbidrag som behandlar om ämnet eMRAM som inbyggd minnestyp för en mikrokontroller. Projektmålet och frågeställningarna besvarades därför genom att en analys utfördes på dokument av olika slag, såsom vetenskapliga artiklar samt vetenskaplig tidskrift från IEEE Xplore och litteraturer samt relevanta webbplatser från KTHB Primo respektive Google Scholar [13]. Genom att använda olika metoder för att uppnå en resultatrik informationssökning samt genom att jämföra all information med flera källor av samma och olika slag för att öka trovärdigheten enligt [13] [14], kunde en sammanställning av informationen från vetenskapliga artiklarna, vetenskapliga tidskrifter och från litteraturen sammanställas. Dessutom kunde en grundlig och en detaljerad bild fås om minnestypen eMRAM och projektmålet samt frågeställningarna kunde därför presenteras och utvärderas utifrån en sammanställning av informationsinsamlingen från litteraturstudien.

D. Rapportformat

I avsnitt II ges en beskrivning på teorin bakom de existerande minnestyperna och för en mikrokontroller, dessutom kommer flera begrepp inom minnesteknologin att presenteras. En översikt av eMRAM introduceras i avsnitt III. I avsnitt IV presenteras resultatet av litteraturstudien. En diskussion kring frågeställningarna utifrån resultatet från avsnitt IV presenteras i avsnitt V. Slutligen kommer en sammanfattning av studien presenteras i avsnitt VI.

II. TEORI

A. Mikrokontroller

En mikrokontrollerenhet är ett datorsystem, som innehåller flera viktiga inbyggda komponenter integrerad i ett och samma chip. De större fundamentala komponenterna i en

generelltyp av mikrokontroller är en processor (CPU), inbyggda minnen, inmatning/utmatningsportar (I/O) och en systembuss [15]. I Fig. 1 visas förutom de fundamentala subsystemen, dessutom på ett flertal komponenter som finns i en mikrokontroller. Dessa komponenter är en klockhanteringsenhet, en beräkningshanteringsenhet (timer samt räknare), ett avbrottmekanism, en kommunikationsmodul (seriekommunikations system), en digital till analog konverterare (DAC) och en analog till digital konverterare (ADC) [15] [4] [3].

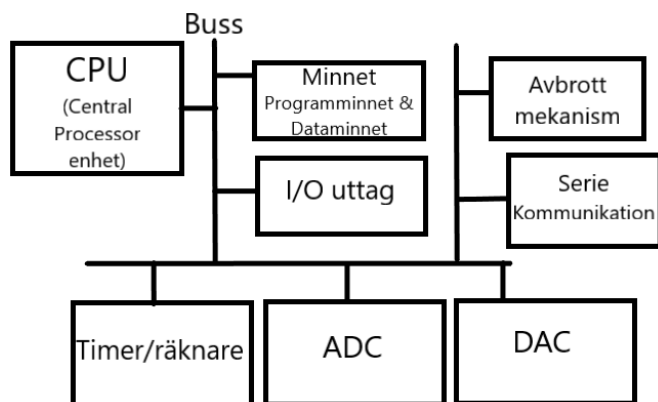


Fig. 1. En generell mikrokontroller arkitektur [4].

1) *CPU*: En CPU är en huvudkontrollenhet för en mikrokontroller vars funktion är att hantera instruktioner som är lagrad i CPU:n och i de inbyggda minnen [15]. CPU utför en programinstruktion i tre steg, först hämtas instruktionerna som är lagrad i minnet och sedan avläses och avkodas instruktionsinnehållet. I sista steget utförs programinstruktionerna, som består av en serie av specifika förprogrammerade uppgifter. En CPU består av en aritmetisk logikenhet (ALU) och en kontrollenhet (CU) som utför programinstruktionerna [4].

2) *Programminnet och dataminnet*: Programminnet och dataminnet är två grundminnestyper för en mikrokontroller minnesarkitektur [15] [4]. Flash och EEPROM, som är programminnet är den delen av minnet som innehåller programinstruktionerna. SRAM, som är dataminnet är den delen av minnet som utför en läsoperation samt en skrivoperation av programinstruktionerna och temporärt lagrar resultaten av instruktionerna under en systemoperation [15] [4] [3]. Flash är en minnestyp som används för att komplettera lagringen av resultaten efter en färdig systemoperation av programinstruktionerna. EEPROM är en minnestyp som generellt används för lagring av resultat, variabler och programinstruktioner i CPU:n [4].

3) *I/O-portar*: I/O möjliggör att en mikrokontroller kan användas av externa enheter [15]. Inmatningsenheterna kommunicerar med en mikrokontroller genom att binära data skickas från enheterna i digitala signaler till CPU:n. I/O portar är generellt åtta bit inmatnings samt utmatningsportar. En I/O port består av ett uttagsregister tillsammans med ett datarikttningsregister [4].

4) *Systembuss*: En systembuss är en grupp av parallella anslutningskablar som ansluter CPU:n till olika subsystem av komponenter i systemet [15]. En mikrokontroller har tre typer av systembussar, adressbuss, databuss och kontrollbuss. En adressbuss är anslutningen mellan CPU och de inbyggda minnen. Adressbussen definierar en mikrokontroller kapacitet av separata adressminnesplatser och kapaciteten för ett system bestäms av antalet anslutningar (i bit) från CPU:n till de inbyggda minnen. En databuss är anslutningen mellan de olika komponenterna i en mikrokontroller. Bredden för en databuss bestämmer storleken av ett dataargument, som en mikrokontroller kan hantera. För en mikrokontroller varierar bredden för en databuss (i bit) från 4-bit till 32-bit. En kontrollbuss är anslutningarna till de olika komponenterna för att skicka och ta emot kontrollsignaler i en mikrokontroller. Kontrollsignalerna används av CPU:n under en systemoperation så att programinstruktionerna kan utföras på ett korrekt sätt [4].

5) *Timer och räknare*: En timer och en räknare kontrollerar tiden för en systemoperation av programinstruktionerna samt tillåter olika operationer, såsom tidsfördröjning och att en mikrokontroller kan utföra systemoperationer i specifika frekvenser [4].

6) *Seriekommunikationssystem*: En seriekommunikation är en kommunikationsteknik som tillåter de olika subsystemen att kommunicera och synkronisera mellan varandra samt de externa enheterna, genom att skicka och ta emot data. Det finns två typer av seriekommunikationssystem för en mikrokontroller, en synkron kommunikation och en asynkron kommunikation. En synkron seriekommunikation använder en synkroniserad klocka för att synkronisera en sändning och en mottagning av databit. En synkron seriekommunikationsmodul är en 16-bit register, åtta-bit register för sändning och åtta-bit register för mottagning. En asynkron seriekommunikations synkronisering av en sändare och en mottagare används en start och stopp bit metod för en sändning och en mottagning av databit i form av åtta-bit signaler [4].

7) *Avbrottmekanismsystem*: Ett avbrottsystem tillåter en mikrokontroller att temporärt stoppa en systemoperation av programinstruktioner, för att utföra ett specifikt systemaktivitet av högre prioritet. Mekanismen utförs i fyra steg och ett system kommer först att färdigställa den aktuella programinstruktionen. En returadress, ett registervärde och informationen av den nästkommande programinstruktionen lagras i en stack. Stacken är ett tillfälligt lagringsplats för avbrottsystemet. Mekanismens avbrott servicerutin (ISR) utför systemaktivitetet som orsakade ett avbrott i systemet. Den lagrade informationen i stacken återanvänds och mikrokontrollen återställs så att en normal systemoperation av programinstruktionerna kan utföras. Det finns två typer av avbrottsystem för en mikrokontroller, ett avbrott i realtid (RTI) och ett externt avbrottförfrågan (IRQ). En RTI utför rutinmässiga periodiska avbrott i en mikrokontroller systemoperation. IRQ är ett externt avbrottsystem som utför ett avbrottmekanism när systemets stift aktiveras av

externa systemenheter [4].

8) *ADC system:* En analog till digital konverterare (ADC) är ett subsystem som konverterar analoga signaler från analoga inmatningsenheter till digitala signaler för CPU [15] [4]. De digitala signalerna konverteras sedan till binära representationer för CPU:n.

9) *DAC system:* En digital till analog konverterare (DAC) är ett subsystem som konverterar digitala signaler från CPU:n i en mikrokontroller till analoga signaler för externa analoga enheter [15].

B. Begrepp inom minnesteknologin

1) *Icke-flyktigt minne:* Ett icke-flyktigt minnestyp inom minnesteknologin definieras för ett minne som kan bevara data och information när spänning kopplas bort från systemet. Generella minnestyper med ett icke-flyktigt egenskap är lagringsminnen, som lagrar data och information för ett system [16].

2) *Flyktigt minne:* Flyktiga minnen inom minnesteknologin är en definition för ett minne som inte kan bevara data och information när spänning kopplas bort. Generella minnestyper med flyktiga egenskaper är läs- och skrivminnen [16].

3) *Minnesuthållighet:* Minnesuthålligheten för ett minne är ett mått på hur många skriv- och raderingscykler som kan utföras av en minnestyp innan ett misslyckande inträffar. Det vill säga antalet programmeringscykler som ett minne kan utföra under en minnesoperation innan minnet misslyckas med att läsa tillbaka korrekt data. Antalet skriv- och raderingscykler, beror på flera faktorer som kan påverka den övergripande minnesuthålligheten hos ett minne. Det är ett minnes driftförhållande, såsom höga temperaturer och en hög spänning som medför en minskad minnesuthållighet [17].

4) *Databevaringstid:* En databevaringstid är ett mått på förmågan att kunna bevara information när spänning kopplas bort från ett minne. Det betyder att ett minnes databevaringstid definierar hur lång tid minnet kan säkerställa att all data kommer att bevaras och beroende på vilken typ av minne, så är icke-flyktiga minnen kapabelt att uppnå en databevaringstid upp till flera tiotals år [17].

5) *Minnesdensitet och minnesarea:* En minnesdensitet definieras som antalet bitceller som får plats i en minnescell. Minnescellen är en matris av bitceller och en minneskapacitet definieras därav i bits. Bitcellerna tillåter ett minne att lagra de binära representationerna av logisk "1" och logisk "0". Densiteten för ett minne beror därav på antalet bitceller och bitcellernas storlek. Ett minne med hög densitet består av flera bitceller i en mindre storlek för en minimal chiparea [18].

En minnesarea är hur mycket av en chiparea som används för att bilda en minnescell. Det definieras som ett mått på minnets fysisk cellstorlek, genom en mätning av bitcellernas storlek och den minsta funktionen som kan skapas. Storleken

mäts därav av en teknikoberoendemetrikt (F^2). F^2 är ett mått på den minimala design avståndet som kan användas i en krets [6]. F^2 är därför den minsta 2D funktionsstrukturen som kan skapas och en funktionsstorlek (F) är detsamma som en tekniknodstorlek. En tekniknod är den fysiska storleken och dimensionen av en transistor och dess storlek är den minimala dimensionen som en CMOS teknikprocess kan skapa [18].

6) *Minnesoperation:* Det finns generellt två typer av minnesoperationer, en läsoperation och en skrivoperation. En minnesoperation tillåter ett minne att lagra de binära data i en bit. Det binära representationen för logisk "1" och logisk "0" under en läsoperation och en skrivoperation, skriver och läser minnet en bitcell eller en grupp av bitceller. En minnesoperation kan variera beroende på minnestypen och bitcellernas arkitektur [19].

C. SRAM

SRAM är en minnesoperations intensiv minnestyp och minnesapplikationen är därav inom flera områden, framför allt som en arbetsminne för en mikrokontroller och som en cacheminne. I Tabell I visas SRAM minnesegenskaper samt energikostnad per bit för varje minnesoperation [6] [3]. En SRAM minnescell är en matris av bitceller som är uppbyggd av flera MOSFETs. Det finns flera typer av SRAM som har olika många transistorer. De vanligaste bitcellarkitekturerna för SRAM minnet är en fyra-transistor (4T) SRAM, en sex-transistor (6T) SRAM, en åtta-transistor (8T) SRAM och en tio-transistor (10T) SRAM. I Fig. 2 visas en generell SRAM bitcellarkitektur som består av sex transistorer. Två av transistorerna är passgrind transistorer som kontrollerar tillgången från bitlinjerna (BL och \overline{BL}) till lagringscellen under en minnesoperation. De inre transistorerna i lagringscellen är två motriktade CMOS inverterare. Transistor ett och transistor två är den första CMOS inverteraren samt transistor tre och transistor fyra är den andra CMOS inverteraren. En minnesbit är lagrad i de två CMOS inverterare, som tillåter att minnet kan inta två tillstånd genom en vippa och lagrar på så sätt de binära tillstånden logisk "1" och logisk "0" i varje bitcell [9] [20].

SRAM har tre minnesoperationer, en läsoperation, en skrivoperation och en vilolägesoperation. I vilolägesoperationen för minnet har ordlinjen (WL) i minnescellen en låg spänningsnivå och medför att båda passgrind transistorerna isolerar de inre två CMOS inverterare från BL och \overline{BL} . SRAM har inte någon uppdateringscykel för att behålla den elektriska laddningen som lagrar de binära tillstånden i bitcellerna. Om en logisk "1" är lagrad i ingången i den första CMOS inverteraren kommer en logisk "0" att vara lagrad i både utgången av den första CMOS inverteraren, men även i ingången av den andra CMOS inverteraren. Det betyder att en logisk "1" är därmed lagrad i utgången av den andra CMOS inverteraren. De binära tillstånden lagrade i SRAM kommer att återkopplas i bitcellerna och vara lagrade fram tills spänningen kopplas bort från minnesenheten [9].

I en läsoperation för minnet har både BL och \overline{BL} en hög spänning och är elektriskt laddade. WL får en

spänning och de två passgrind transistorerna aktiveras. De två spänningsskillnaderna lagrade i de inre transistorerna i bitcellen leder till en minskad spänning i BL och \overline{BL} , som motsvarar spänningen i passgrind transistorerna. BL har då en spänningsnivå som passgrind transistor ett och \overline{BL} har en spänningsnivå som passgrind transistor två. En avkänningsförstärkare (SA) detekterar spänningsskillnaden mellan BL och \overline{BL} och den lagrade binära tillståndet i bitcellen kan avläsas under minnesoperationen.

Tabell I
EN ÖVERSIKT ÖVER MINNESEGNSKAPER FÖR SRAM

Minnestyp	SRAM
Fysisk cellstorlek (area i F^2)	150-200 F^2
Chiparea (1Mb)	204 000 μm^2
Densitet [bit] (kapacitet)	<1G-bit
Max. antal programmeringscykler	$>10^{10}$ (∞)
Programmeringstid (skrivoperationstid)	5-40ns
Åtkomsttid (Läsoperationstid)	1-20ns
Databevaringstid	0
Läsenergi (pJ/bit)	0.96 (pJ/bit)
Skrivenergi (pJ/bit)	0.73 (pJ/bit)

I en skrivoperation har både BL och \overline{BL} en spänning som motsvarar de binära tillstånden logisk "1" respektive logisk "0". En skrivoperation för logisk "1" har BL samt den andra CMOS inverteraren en spänning som motsvarar logisk "1" och \overline{BL} samt den första CMOS inverteraren har en spänning som motsvarar logisk "0". WL har en hög spänning och det leder till att de två passgrind transistorerna aktiveras. Spänningen i BL passerar den första inre CMOS inverterarens ingång som motsvarar en logisk "1" och samtidigt passerar spänningen i \overline{BL} den andra inre CMOS inverterarens ingång som motsvarar en logisk "0". Det binära tillståndet logisk "1" är därmed lagrad i bitcellen och för en skrivoperation för logisk "0" sker det genom en omvänd process och på så sätt kan det binära tillståndet logisk "0" lagras i bitcellerna [9] [20] [18].

Storleken av SRAM minnet bestäms av storleken av minnescellen. Minnescellstorleken definieras därför som 2^m ord eller som $2^m \times n$ bits, där m är antalet adresslinjer (bitlinje) och n är antalet datalinjer (ordlinje).

D. Flash

Det finns två typer av flashminnen, en NOR flash och en NAND flash [6] [21] [22]. Flashminnescellen är uppbyggd av en matris av bitceller och varje bitcell har en bitcellarkitektur, som kallas för en flytandegrind MOSFET (FGMOSFET). Bitcellernas FGMOSFET är en två grindtransistor, som består av en kontrollgrind (CG) och en flytandegrind (FG). I Fig. 3 visas en generell flashminnes bitcellarkitektur som består av en transistor, en FG, en CG, en WL och en BL. WL är kopplad till CG, som har ett isolerande oxidlager på undersidan. FG är placerad under CG samt ovanpå transistor och har två oxidlager som är placerade ovanpå och på undersidan av FG.

Det tillåter att en elektrisk laddning i FG kan lagras och representera de två binära tillstånden logisk "1" och logisk "0" genom två elektriska laddningstillstånd i FG, som motsvarar att det finns en elektrisk laddning respektive att det inte finns någon elektrisk laddning [21] [8] [22].

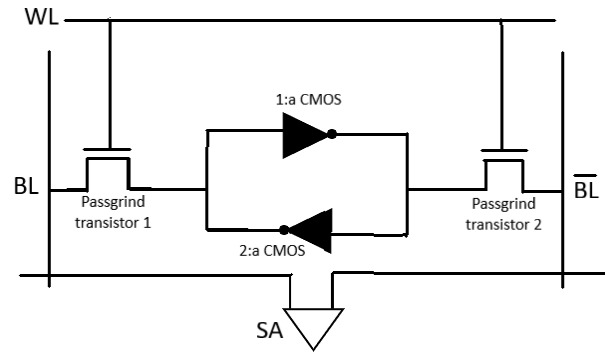


Fig. 2. En 6T-SRAM bitcellarkitektur [9] [20].

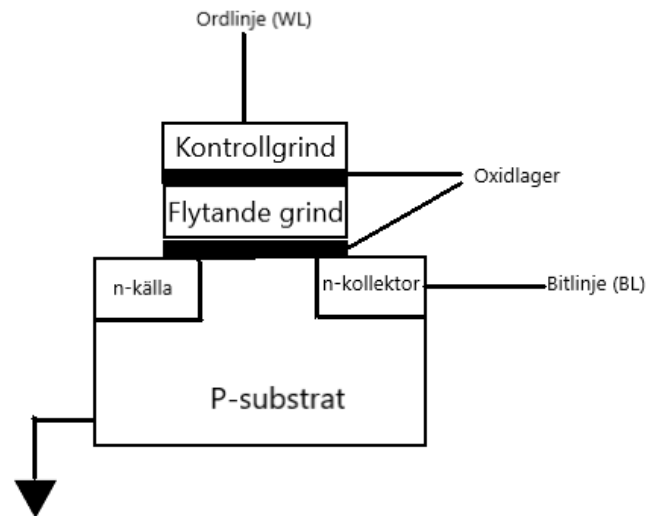


Fig. 3. Ett flashminnets FGMOSFET bitcellarkitektur [22]

Ett flashminne har tre minnesoperationer, en läsoperation, en skrivoperation och en raderingsoperation. Minnesoperationerna kan beroende på flashminnestypen utföras i blockvis operationer. Bitcellerna i en flashminnescell är organiserad i strängar, som beroende på typen av flashminnet antingen består av seriekopplade eller parallellkopplade bitceller. En sträng av bitceller är sedan organiserad i sidor som består av flera strängar och sedan i ett block som består av flera sidor av bitcellsträngar.

Det finns tre typer av minnescelllagring för ett flashminne, en-nivå-cell (SLC), fler-nivå-cell (MLC) och en trippel-nivå-cell (TLC) [23] [21]. En SLC flashminnescell kan lagra en bit i varje minnescell och en SLC representerar de två binära tillstånden genom två elektriska laddningstillstånd i bitcellerna. Det första elektriska laddningstillståndet är att FG inte har en elektrisk laddning och det representerar en logisk "1" som är lagrad i bitcellen. Det andra elektriska

laddningstillståndet är att FG har istället ett flöde av elektroner, som ger en negativ laddning i grunden. Det representerar att en logisk "0" är lagrad i bitcellen. I en MLC flashminnescell lagras två bits i varje minnescell och MLC har därför fyra elektriska laddningstillståndsnivåer för att representera de binära tillstånden i bitcellerna. De binära tillstånden logisk "00", logisk "01", logisk "10" och logisk "11" representeras i en MLC flashminnescell genom att nivån av den elektriska laddningen (flödet av elektronerna) detekteras i FG. I en TLC flashminnescell lagras tre bits i varje minnescell och en TLC har därför åtta elektriska laddningstillståndsnivåer för att representera de binära tillstånden i bitcellerna [23].

Tabell II
EN ÖVERSIKT ÖVER MINNESEGNSKAPER FÖR NOR-FLASH

Minnestyp	NOR-flash
Fysisk cellstorlek (area i F^2)	8-10 F^2
Densitet [bit] (kapacitet)	<16G-bit
Max. antal programmeringscykler	10^4 - 10^5
Programmeringstid (skrivoperationstid)	5-10 μs
Åtkomsttid (Läsoperationstid)	slumpmässig: 80-150ns, serie: 80-120ns
Databevaringstid	>10år

1) *NOR-flash*: En NOR flashminnescell består av flera bitceller som har den generella bitcellarkitekturen för en flashminnescell. Bitcellerna i en NOR flash är parallellkopplade med varandra och varje ände av bitcellerna är kopplade till en BL och en jord. Denna serie av FG transistorer bildar en matris av NOR flashminnesceller.

En raderingsoperation för NOR flashminnet sker blockvis och en läs- och skrivoperation sker på åtta bits (bytes). I Tabell II visas NOR flash minnesegenskaper [6] [3]. I en läsoperation för minnet får en bitcells CG en hög spänning genom WL och de resterande bitcellerna i den parallellkopplade serien sätts CG till en låg spänning genom respektive WL. Om FG har en elektrisk laddning är bitcellens binära tillstånd en logisk "0" respektive en logisk "1" om FG inte har en elektrisk laddning.

I en skrivoperation för NOR flash används injektion av heta elektroner som skrivoperationsmetod. BL har en hög spänning och en bitcells CG får därmed en hög spänning, samtidigt sätts en låg spänning i de resterande bitcellernas CG i minnescellen och ett elektronflöde uppstår mellan två terminaler i bitcellens transistor. Elektronflödet i transistoren är hög och passerar den isolerande oxidlagret mellan FG och transistoren. FG har en elektrisk laddning och tillståndet representerar en logisk "0" i bitcellen. I en raderingsoperation för NOR flash tillförs en negativ spänning till CG i transistoren för alla bitceller i ett block av en flashminnescell och det gör att elektronerna i FG stöts bort av den negativa elektriska laddningen från CG. FG töms av sitt elektronflöde och det binära tillståndet i bitcellerna för ett block representerar en logisk "1" [21] [22].

2) *NAND-flash*: En NAND flashminnescell består av flera bitceller som har den generella bitcellarkitekturen för en flashminnescell. Bitcellerna i en NAND flash är seriekopplade

med varandra, där ena terminalen av transistoren för en bitcell är seriekopplad med den andra terminalen av en transistor för en annan bitcell. I serien är den sista bitcellens ena terminal seriekopplad med bitlinjetransistorns motsatta terminal och den andra änden av bitcell serien är seriekopplad med den motsatta terminalen av transistoren för jordning. Bitlinjetransistorns andra terminal är kopplad till en separat BL [21] [22] [23].

Tabell III
EN ÖVERSIKT ÖVER MINNESEGNSKAPER FÖR NAND-FLASH

Minnestyp	NAND-flash
Fysisk cellstorlek (area i F^2)	4-5 F^2
Densitet [bit] (kapacitet)	<512G-bit
Max. antal programmeringscykler	10^3 - 10^4
Programmeringstid (skrivoperationstid)	100-300 μs
Åtkomsttid (Läsoperationstid)	slumpmässig: 10-20 μs , serie: 5-50ns
Databevaringstid	>10år
Läsenergi (pJ/bit)	1.23 (pJ/bit)

En läs- och skrivoperation för en NAND flash sker per sida och en raderingsoperation sker blockvis. I Tabell III visas NAND flash minnesegenskaper [6]. I en läsoperation för minnet får bitcellens CG en hög spänning genom WL, dessutom får de resterande bitcellerna en hög spänning i den seriekopplade NAND flash minnescellstrukturen. Om FG har en elektrisk laddning är bitcellens binära tillstånd en logisk "0" respektive en logisk "1" om FG inte har en elektrisk laddning.

I en skrivoperation för NAND flash används tunnelinjektion av elektroner som skrivoperationsmetod. En hög spänning tillförs till bitcellens WL och CG har då en hög spänning. Ett elektronflöde uppstår mellan terminalerna i transistoren, som passerar oxidlagret mellan FG och transistoren. Det medför att FG har en elektrisk laddning och tillståndet representerar en logisk "0" i bitcellen. I en raderingsoperation för NAND flash har minnestypen en liknande raderingsminnesoperation som NOR flashminnet. Det binära tillståndet i bitcellerna i ett block representerar en logisk "1" efter en raderingsoperation [21] [23].

E. EEPROM

En EEPROM minnescell är likt en flashminnescell, som är uppbyggd av en matris av bitceller. Bitcellarkitekturen består av en FG tunneloxid transistor (FLOTOX) [24] [25]. I Tabell IV visas EEPROM minnesegenskaper, som till följd av minnets minnescellarkitektur [6] [26] [27] [24]. I Fig. 4 visas en EEPROM bitcellarkitektur av en FLOTOX som består av två transistorer, en FGMOSFET och en minnesoperations transistor. Den första transistoren är en lagringstransistor för EEPROM minnet. Den andra transistoren kontrolleras av WL, genom att en hög spänning tillförs till WL och transistoren sätts på under minnesoperationerna. Det innebär ett minskat tillförd spänning till FGMOSFET vars FG isoleras av ett tunt oxidlager. Ett binärt tillstånd definieras därav genom två

elektriska laddningstillstånd, om FG i den första transistorn har en elektrisk laddning motsvarar det en logisk "1" lagrad i bitcellerna respektive om den inte har en elektrisk laddning motsvarar det en logisk "0".

EEPROM har tre minnesoperationer, en läsoperation, en skrivoperation, en raderingsoperation och minnesoperationerna sker i bytes för minnestypen. I en läsoperation för EEPROM har BL, WL och CG en hög spänning. En representation av logisk "0" i BL har FG inte någon elektrisk laddning och det medför en låg tröskelspänning samt ett elektronflöde i transistorns terminaler. En representation av logisk "1" i BL har FG en elektrisk laddning, som då innebär en hög tröskelspänning och att det inte uppstår ett elektronflöde i transistorns terminaler.

Tabell IV
EN ÖVERSIKT ÖVER MINNESEGEGNSKAPER FÖR EEPROM

Minnestyp	EEPROM
Fysisk cellstorlek (area i F^2)	$4F^2$
Fysisk cellstorlek	$>50\mu m^2$
Densitet [bit] (kapacitet)	$<2M\text{-bit}$
Max. antal programmeringscykler	$10^4\text{-}10^5$
Programmeringstid (skrivoperationstid)	$5\text{-}80ms$
Åtkomsttid (Läsoperationstid)	$5\text{-}20ms$
Databevaringstid	$>10\text{år}$

I en skrivoperation för minnet används tunnelinjektion av elektroner, som då representerar logisk "0" respektive används en omvänd process av skrivoperationsmetoden för logisk "1". WL samt BL har en hög spänning och CG har en låg spänning om minnet utför en skrivoperation för en logisk "0". Transistorns emitter terminal och kollektor terminal har inte något strömflöde och ett positivt elektriskfält attraherar då elektronflödet i FG, som passerar genom ett tunt oxidlager och dras till transistorns kollektor terminal. FG har därför inte en elektrisk laddning och tillståndet representerar en logisk "0". Om CG istället har en hög spänning samt att transistorns terminaler, emitter och kollektor har ett elektronflöde, utför minnet då en skrivoperation för en logisk "1". Den elektriska laddningen i CG attraherar därför elektronerna från transistorns terminaler till FG, som medför att det finns en elektrisk laddning.

I en raderingsoperation återställs bitcellerna antingen till logisk "0" eller till logisk "1", beroende på om det finns en elektrisk laddning i FG innan en skrivoperation har utförts. En raderingsoperation från det binära tillståndet logisk "0" till logisk "1" har CG och WL en hög spänning. Ett positivt elektriskfält uppstår i CG och attraherar elektronerna från transistorns terminaler till FG. FG har då en elektrisk laddning och bitcellen återställs till det tidigare binära tillståndet. En omvänd process tillämpas för en raderingsoperation från logisk "1" till logisk "0" [25] [28] [26].

III. EMRAM

A. eMRAM översikt

eMRAM har som egenskap att kunna användas som en lagringminne för kodinstruktionsdata (icke-flyktigt minnesegenskap) och som en arbetsminne (flyktigt minnesegenskap) [29] [3]. I Tabell V och enligt [9] [30] [29] beskrivs det att eMRAM har en obegränsad minnesuthållighet, samtidigt som minnestypen har en minnesdensitet kapacitet som uppnås till 1G-bit. Minnestypens icke-flyktiga egenskaper beskrivs i [30] [29] att det är en konsekvens av att eMRAM har databevaringstids egenskaper som är >10 år och samtidigt uppfyller applikationskrav för industriella temperaturer [29], som för eMRAM har en drifttemperaturintervall på $-40\text{-}150^\circ C$.

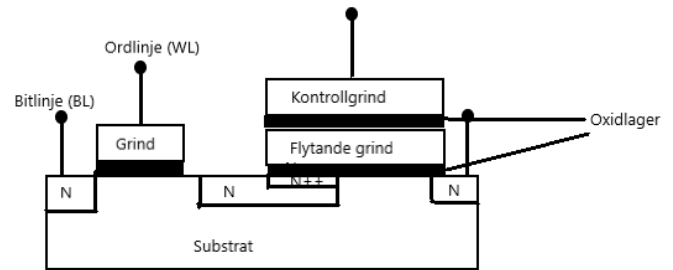


Fig. 4. En EEPROM FLOTOX bitcellarkitektur [24]

Det framgår i [9] [29] att eMRAM är en byte adress-bar minnestyp som tillåter bitvis alternering, vilket ger som följd att eMRAM har snabba skriv- och läsoperationstider på $<10ns$ respektive $<5ns$. En snabb minnesoperationstid samtidigt som den har en hög minnesuthållighet, betyder det att eMRAM uppnår en minnesprestanda som är likt slumpmässigt åtkomstminnen. Vidare beskrivs det i [9] att eMRAM har en minnescellstruktur som använder en transistor och en magnetoresistiv stackstruktur. Det betyder att eMRAM har en cellstorlek area mellan $16 - 50F^2$ för en given tekniknod [3] [31] och som innebär att eMRAM är ett skalbart minne. Det betyder att minnet kan skalas ner i de lägre tekniknoderna för ett inbyggt system, samtidigt som det inte leder till att cellstorlek arean ökas till de högre F^2 nivåerna. Minnets icke-flyktiga egenskaper innebär även att eMRAM som minnestyp inte har något strömläckage under drifttid. Egenskapen beror på att det endast uppstår ett strömflöde för minnet under en minnesoperation samt att den använder magnetisk moment för att lagra element, därav krävs det inte en konstant spänning för att lagra och bevara data i eMRAM [32]. I Tabell V visas det även att minnestypen har en låg energikostnad för skriv- och läsenergi på <200 (fJ/bit) respektive <20 (fJ/bit).

B. MTJ-tekniken

En MTJ stack består av tre huvudkomponenter i form av magnetiska och icke magnetiska lager, som innebär att konfigurationstekniken fungerar som ett magnetiskt lagringselement för att lagra de binära tillstånden för minnet. De tre lagren består enligt [9] och [30] av två ferromagnetiska metall lager (CoFeB) och mellan dessa två metall lagren finns ett tunt isolerande lager ($\leq 1nm$ [9] eller $0.85nm$ [33]), som kallas

för oxidlagret och som är av typen magnesiumoxid (MgO). I Fig. 5 fungerar oxidlagret som en tunnelbarriär för MTJ stacken, som ett elektronflöde passerar igenom. Dessutom fungerar den som en isolerande barriär, vilket separerar de två ferromagnetiska lagren. Det ger en följande MTJ stackstruktur, CoFeB/MgO/CoFeB och det beskrivs i [9] [33] [30] att de två ferromagnetiska lagren är lagringslagret (frilager) samt referenslagret (RL).

Tabell V
eMRAM MINNESEGNSKAPER

Minnestyp	eMRAM
Fysisk cellstorlek (area i F ²)	16-50 F ²
Densitet [bit] (kapacitet)	<1G-bit
Max. antal programmeringscykler	>10 ¹⁰ (∞)
Programmeringstid (skrivoperationstid)	5-10ns
Åtkomsttid (Läsoperationstid)	1-5ns
Databevaringstid	>10år
Läsenergi (fJ/bit)	10-20
Skrivenergi (fJ/bit)	100-200
Max. drifttemperatur	-40-150°C

Ovanpå den ferromagnetiska frilager (FL) finns ett täckande lager, som tillåter justering av magnetiska egenskaper för den ferromagnetiska FL och fungerar som ett skydd mellan den övre elektroden och FL. Den ferromagnetiska RL består av två olika ferromagnetiska fastlager (första fastlagret samt andra fastlagret), som mellan dessa finns det placerat ett icke ferromagnetisk lager (Ru). Det ger en följande struktur för RL, CoFeB/Ru/CoFeB. Den första och den andra fastlagret har fasta magnetiska polarisationsriktningar åt respektive riktning. Det representeras av enkelriktningen på pilarna, som har motsatta riktningar [9]. Det betyder att RL har en polarisationsriktning som alltid är fast i en magnetisk orientering och att ett strömflöde som passerar genom RL inte ändrar orienteringen [30]. En sådan egenskap beskrivs i [9] att det uppnås av den syntetiska anti-ferromagnetiska (SAF) upplägget, som består av en struktur med RL placerad ovanpå en anti-ferromagnetisk fastlager.

FL har istället en varierande polarisationsriktning, som tillåter att den magnetiska orienteringen varierar i förhållande till RL polarisationsriktning och representeras i FL av den dubbelriktade pilen. Det innebär att ett strömflöde som passerar genom FL kommer att ändra lagrets magnetiska orientering [30] och en MTJ stack kan därav representera två magnetiska energitillstånd [33]. Om FL har en magnetisk polarisation i samma riktning som RL magnetiska orientering, befinner sig MTJ i ett parallellt (P) tillstånd. Om FL istället har en polarisation i en motsatt riktning till RL polarisationsriktning, befinner sig MTJ i ett anti-parallellt (AP) tillstånd [30] [34].

C. Magnetisk tunnelresistans

En magnetisk tunnelresistans (TMR) är en effekt som uppstår i MTJ när konfigurationstekniken för eMRAM befinner sig i de två energitillstånden, P och AP. Nivån av MTJ resistansen representeras enligt [34] av två diskreta MTJ resistansvärden, en låg resistans (R_L) och en hög resistans (R_H). Det korresponderar att de två MTJ energitillstånden och MTJ resistanserna representerar en låg resistans för P tillståndet (R_P) och en hög resistans för AP tillståndet (R_{AP}) [30]. TMR representerar amplituden av motståndsförändringen och det innebär att strömflödet genom MTJ stacken varierar i de två energitillstånden [34]. När det är en låg resistans i ett P tillstånd för MTJ, medför det att strömflödet enkelt passerar genom oxidlagret till antingen FL eller RL och det innebär att det finns ett högt strömflöde genom MTJ stacken. Om de två ferromagnetiska lagren i MTJ har två olika polarisationsriktningar, dvs när det är en hög resistans och MTJ stacken befinner sig i ett AP tillstånd. Det innebär att strömflödet har det svårare att passera genom MTJ och därav finns det ett lågt strömflöde genom MTJ stacken [9].

TMR är en viktig faktor för en eMRAM minnesoperation, då de två tillstånden (R_{AP} och R_P) representerar en logisk "1" respektive en logisk "0" [30]. Effekten används framför allt för en eMRAM läsoperation och en hög TMR effekt innebär en förbättrad läsoperationstid, eftersom effekten används för att mäta skillnaden mellan MTJ cell resistansen och den fördefinierade referenscell resistansen genom MTJ stacken, se Fig. 6 [9].

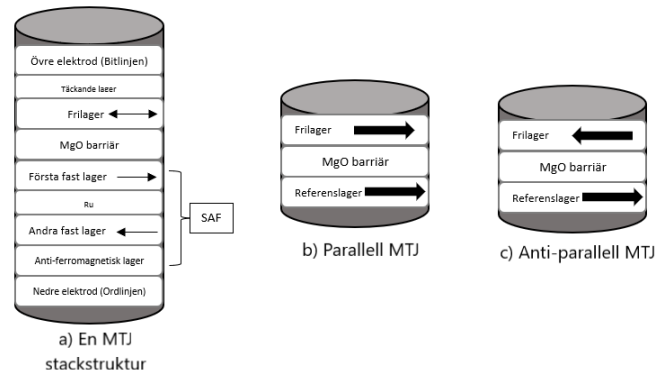


Fig. 5. En generell MTJ stackstruktur och en MTJ i en parallell (P) tillstånd respektive i en anti-parallell (AP) tillstånd [9].

D. STT-mekanismen

I STT-mekanismen används spinnberoende spridning av elektroner för att överföra det magnetiska momentet och därmed ändra magnetiseringsstillståndet av ferromagnetiska FL i MTJ. STT effekten tillåter att MTJ kan då byta mellan de två energitillstånden (P till AP eller AP till P). För ett byte från AP till P har den ferromagnetiska FL en magnetiseringsriktning anti-parallell till RL. Det beskrivs i [30] att ett dubbelriktat strömflöde av fria elektroner från den nedre elektroden har ett spinn av elektroner, som genererar magnetiskt moment i en riktning som antingen har en uppåtgående spinn eller nedåtgående spinn. En majoritet av det fria elektronflödet

passerar då genom den ferromagnetiska fastlagret och elektronerna får en polarisation likt den magnetiska riktningen för fastlagret. Det leder till enligt [9] att det spinnpolariserade elektronflödet passerar genom oxidbarriären och överför en magnetisk moment till FL. Om elektronflödet är tillräckligt hög för att uppnå den kritiska växlingsströmtröskel (I_C), roteras FL polarisationsriktning från den tidigare AP tillståndet till ett P tillstånd. En minoritet av det spinnpolariserade elektronflödet reflekteras tillbaka i oxidbarriären i MTJ stacken och överför en magnetisk moment till fastlagret. SAF upplägget i RL tillåter att polarisationsriktningen inte ändras av den reflekterande elektronflödet.

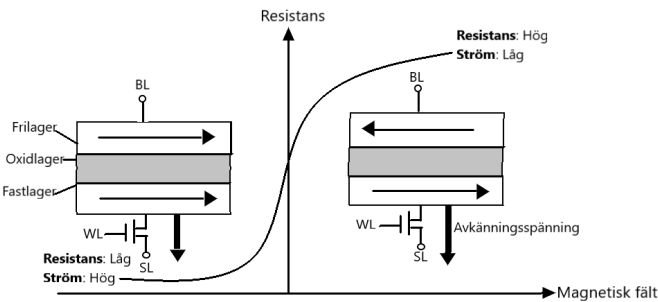


Fig. 6. TMR effekten används för att mäta skillnaden mellan MTJ cell resistansen och den fördefinierad referenscell resistansen genom en MTJ stack. En logisk "1" representeras av R_H och en logisk "0" representeras av R_L . [9].

För ett byte från P till AP beskrivs att det på samma sätt passerar ett dubbelriktat strömflöde av fria elektroner från den övre elektroden till FL [30]. Elektronerna får en polarisation likt den magnetiska riktningen till FL. Då FL befinner sig i ett P tillstånd i jämförelse med RL, passerar en majoritet av elektronflödet genom oxidbarriären. En minoritet av de magnetiska polariserade elektronerna reflekteras från oxidbarriären tillbaka till FL i MTJ stacken och överför en motsatt magnetisk polariserad moment. Elektronflödet roterar FL polarisationsriktning från det tidigare P tillståndet till ett AP tillstånd [9].

E. Minnescellarkitektur och minnesoperation

Det finns två typer av bitcellarkitekturer för en STT-MRAM minnescell, en 1T-1MTJ bitcellarkitektur och en 2T-2MTJ bitcellarkitektur [35]. I Fig. 7 visas den första cellstrukturen som består av en MTJ seriekopplad med en NMOS transistor (1T-1MTJ) [9]. Den andra typen av bitcellarkituren är en cellstruktur som består av en serie av två olika MTJ. Den första MTJ stacken är seriekopplad med en NMOS transistor och den andra MTJ stacken är seriekopplad med en PMOS transistor (2T-2MTJ) [34]. De två minnesoperationerna, en läsoperation och en skrivoperation för en STT-MRAM används STT mekanismen, för att definiera de binära tillstånden logisk "1" och logisk "0" [30].

En läsoperation för minnet med en 1T-1MTJ bitcellstruktur sker genom att en vald bitcell WL sätts till en etta. En läsförskjutningsström tillämpas på antingen den valda cellens BL eller på källlinjen (SL). Om en läsförskjutning

tillämpas på cellens BL sätts SL till jordning och om en läsförskjutning istället tillämpas på cellens SL, då är BL jordad. En avkänningsspänning passerar genom minnets 1T-1MTJ bitcellstruktur och samtidigt som TMR effekten tillämpas kan en läsoperation utföras av minnet. Genom en mätning av resistansen i MTJ stacken kan energitillståndet bestämmas och en detekteringsförstärkare (SA) upptäcker skillnaden mellan cellresistansen och den fördefinierade referenscell resistansen. Om resistansen är låg definieras det som en logisk "0" respektive om den är hög definieras det som en logisk "1". SA bestämmer därefter bitcellens data efter en läsoperation, genom att omvandla signalen från MTJ och producerar en låg spänning (logisk "0") respektive en hög spänning (logisk "1"). SA skapar därmed en riktig nolla respektive en etta med hjälp av spänningsskillnaden.

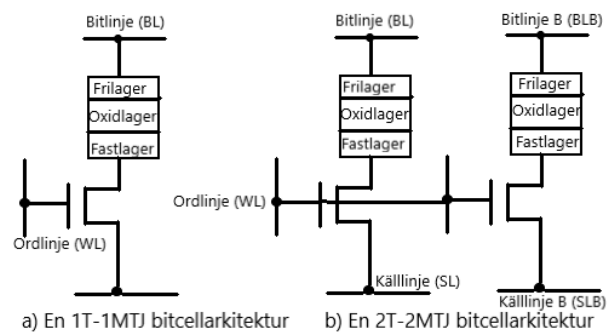


Fig. 7. En 1T-1MTJ bitcellarkitektur och en 2T-2MTJ bitcellarkitektur för en STT-MRAM [9] [35]

I en skrivoperation för eMRAM exploateras elektronernas flöde genom MTJ stacken. I Fig. 8 visas en skrivoperation för det binära tillståndet logisk "1" i en eMRAM minnescell. Ett skrivströmflöde av elektroner passerar genom MTJ stacken från den ferromagnetiska RL och magnetiseringsriktningen för FL är i ett AP tillstånd. Det betyder att RL är seriekopplad med transistorn i 1T-1MTJ. Den ferromagnetiska FL har en majoritet av de spinnpolariserade elektronerna som passerar genom oxidlagret och överför en magnetisk moment som roterar FL polarisationsriktning till ett P tillstånd. Det korresponderar i 1T-1MTJ bitcellen att ett skrivströmflöde har en riktning från BL och SL är därmed jordad. I Fig. 9 visas en skrivoperation för det binära tillståndet logisk "0" i en eMRAM minnescell. Då passerar istället ett skrivströmflöde från FL i MTJ stacken och magnetiseringsstillståndet för den ferromagnetiska FL är i ett P tillstånd och FL är därav seriekopplad med transistorn. En majoritet av de spinnpolariserade elektronerna passerar oxidlagret och en minoritet reflekteras tillbaka till FL vid gränsen till oxidlagret. Det innebär att ett motsatt polariserad elektronflöde roterar FL magnetisering nedåt och polarisationsriktningen för FL blir anti-parallell till RL. Det korresponderar i 1T-1MTJ bitcellen att ett skrivströmflöde har en riktning från SL och BL är därmed jordad [9].

F. Energibarriär och skalbarhet

I [9] och [30] beskrivs det att en STT-MRAM är en skalbar minnestyp. Det framgår att det möjliggör en mindre

formfaktor, en lägre energiförbrukning och därmed minskas tillverkningskostnaderna för eMRAM. Vidare beskrivs det i [9] att en STT-MRAM skalbarhet och minnestypens MTJ stack möjliggör för en mindre I_c , eftersom I_c är dess densitet (J_c) multiplicerat med MTJ area. Det beskrivs i [9] [36] att en skalning av storleken för MTJ stacken ger som följd en mindre I_c , som innebär en minskad skrivenergiförbrukning. En sådan skalbarhet för MTJ beskrivs i [9] att det blir problematiskt att uppnå en hög energibarriär (E_b).

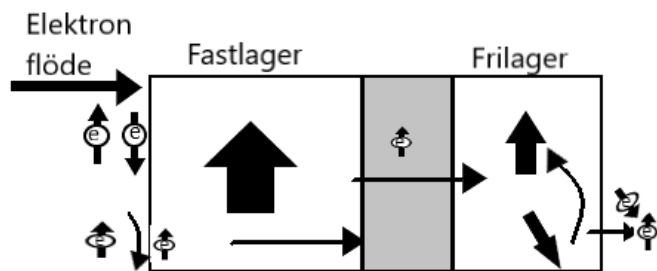


Fig. 8. En skrivoperation för logisk "1" med STT mekanismen för en eMRAM [9].

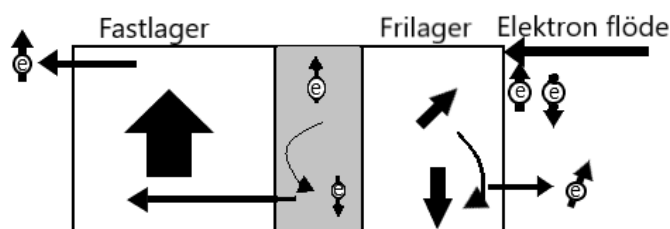


Fig. 9. En skrivoperation för logisk "0" med STT mekanismen för en eMRAM [9].

En E_b är en viktig faktor för en STT-MRAM [9]. Det framgår i [36] att en E_b definierar energin som krävs för att ändra en MTJ magnetiska moment i FL från sin magnetiska orientering. Det magnetiska momentet som uppstår mellan FL och RL, tillåter att en MTJ kan befinna sig i sina två energitillstånd. En E_b är därav energistorleken mellan de två magnetiska energitillstånden, P och AP [33]. Därför är storleken av en E_b proportionell mot eMRAM databevaringstids egenskaper. Vidare framgår det i [9] att en vinkelrät MTJ (peMTJ) för en STT-MRAM tillåter att en hög E_b kan uppnås samtidigt som minnet uppnår en förbättrad skalbarhet. Då förbättrade E_b egenskaper uppnås från en förbättrad materialteknik i MTJ och inte av MTJ formstruktur.

G. Begränsningar med STT-MRAM

De begränsningar som kan uppstå med STT mekanismen ligger till grund på tillförlitligheten av en eMRAM. Det beskrivs i [9] att elektronflödet för en läsoperation (läsströmmen) och elektronflödet för en skrivoperation (skrivströmmen) är parallella med varandra. En sådan egenskap kan som konsekvens medföra att det kan uppstå en läsströmstyrning under en läsoperation för minnet. Det kan dessutom enligt [30] ge som följd av oväntade skrivoperationer när en läsoperation ska utföras. Det framgår att det

beror på STT mekanismen som tillåter samma strömflödesväg för en skriv- och läsoperation. Ytterligare en begränsning med STT mekanismen är att växlingsströmflödet från STT behöver ständigt passera oxidlagret i MTJ stacken och det kan innebära en långsam skrivoperationstid [9]. Det betyder att en hög skrivström är ett krav och då innebär det en ökad skrivenergiförbrukning. Dessutom beskrivs det att ett högt strömflöde krävs för att ändra det magnetiska tillståndet i MTJ från P till AP (i jämförelse med AP till P) [30]. En sådan asymmetri innebär att en hög skrivström behöver tillämpas för att växlingsströmflödet ska ändra tillståndet i MTJ. Det innebär att oxidlagret kan tillslut brytas ner, som en konsekvens av att en hög skrivström ständigt passerar genom lagret i MTJ stacken [9].

IV. RESULTAT

A. Optimering av MTJ stack med peMTJ

Det framgår i [36] [12] att en optimering av MTJ stackstrukturen är ett krav för att STT-MRAM ska kunna tillämpas som både en flash-typ och en SRAM-typ applikation. Det uppges i [36] att en optimering utförs genom att ett ytterligare oxidlager placeras ovanför FL. I Fig. 10 visas en MTJ stack som har ett oxidlager ovanpå FL samtidigt som det finns ett oxidlager på undersidan av FL. Det ger en förbättrad materialteknik för stackstrukturen och en sådan MTJ stack kallas för en vinkelrät MTJ (peMTJ).

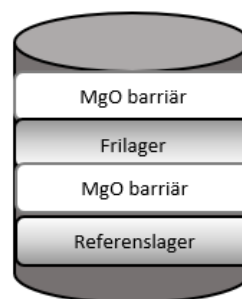


Fig. 10. En optimerad peMTJ, genom att två oxidlager har placerats ovanpå och under frilagret.

Tabell VI
AVVÄGNING MELLAN MINNESUTHÅLLIGHET, DATABEVARINGSTID OCH STRÖMFLÖDET

MTJ stack	MTJ stack A (ej optimerad)	MTJ stack B (optimerad)
Lågt strömflöde från STT	0.83 (a.u)	0.78 (a.u)
Hög minnesuthållighet	Stack B > Stack A	Stack B > Stack A
Högt strömflöde från STT	1.2 (a.u)	0.95 (a.u)
Hög databevaringstid	Stack B > Stack A	Stack B > Stack A

En peMTJ ger dessutom en hög E_b mellan de två energitillstånden, P tillståndet och AP tillståndet för MTJ stacken. Vidare beskrivs det att en hög E_b innebär en högre tröskelspänning för en nedbrytning av MTJ stacken

samt ett kontrollerat växlingsströmflöde av den dubbelriktade strömmen från elektronflödet i STT mekanismen. I Tabell VI visas resultat från [36] på hur en effektiviserad och en kontrollerad växlingsströmflöde från STT mekanismen med en peMTJ ger förbättrade egenskaper för minnesuthålligheten samt databevaringstiden. Det visas att en hög minnesuthållighet är proportionell mot ett lågt växlingsströmflöde från STT mekanismen, då den optimerade MTJ stacken har ett strömflöde på 0.78 (a.u). Dessutom visas det att en hög databevaringstid är proportionell mot ett högt växlingsströmflöde, då den optimerade MTJ stacken har ett strömflöde på 0.95 (a.u).

Tabell VII
MINNESSPECIFIKATIONSKRAV FÖR FLASH-TYP APPLIKATION

Minnestyp	Flash
Minnets chipstorlek	Mindre än inbyggd flashminnets chipstorlek
Uthållighet (antal programmeringscykler)	10^5 - 10^6
Databevaringstid	>10 år (drifttemperatur på 85 - $150^\circ C$)
Dataminnestyp	Icke-flyktigt minne
Läsoperationstid	$<50ns$
Skrivoperationstid	$<1\mu s$
Energi (pJ/bit)	$\leq 1pJ/bit$
Bitcellarkitektur	1T-1MTJ

B. Minnesspecifikationskrav för flash-typ applikation

I Tabell VII visas minnesspecifikationskrav från [36] [12] för en STT-MRAM som en flash-typ applikation. Det visar att en hög databevaringstid på >10 år samt att minnet klarar av drifttemperaturer på 85 - $150^\circ C$ är ett krav. Det framgår att en hög databevaringstid uppnås genom en peMTJ stack och det betyder en kontroll över strömflödet från STT mekanismen i MTJ stacken. Vidare uppges det att som en konsekvens av en hög databevaringstid kommer växlingsströmflöde från STT mekanismen att vara hög. Det beskrivs vidare i [36] att en hög växlingsströmflöde ger som följd en försämrad minnesuthållighet och en hög skrivenergiförbrukning. I [12] framgår det dock att E_b förbättras med en optimerad MTJ stack och att den då har en låg nivå av misslyckande för en minnesuthållighet upp till 50×10^6 cykler. Tabell VII visar dessutom på att för en flash-typ applikation av en STT-MRAM är kravet på minnesuthålligheten lågt, som visar ett maximalt antal programmeringscykler på 10^5 - 10^6 cykler, samtidigt som kravet på skrivoperationstiden är upp till en mikrosekund.

C. Minnesspecifikationskrav för SRAM-typ applikation

I Tabell VIII visas minnesspecifikationskrav från [36] [12] för en STT-MRAM som en SRAM-typ applikation. Det visar att en hög minnesuthållighet på $>10^{10}$ cykler, samt en snabb läsoperationstid och skrivoperationstid på $<50ns$ är ett krav för en SRAM-typ applikation. Vidare uppges det att en

hög minnesuthållighet uppnås genom en peMTJ stack. Det beskrivs vidare i [12] att som en konsekvens av en hög minnesuthållighet och en snabb läs- och skrivoperationstid, ger det som följd ett lågt växlingsströmflöde från STT mekanismen och skrivströmstyrningen respektive läsströmstyrningen ökas. I [36] framgår det hur läsströmstyrningen och skrivströmstyrningen påverkas av ett lågt växlingsströmflöde. Det visar sig att läsströmstyrningen är låg efter 10^6 läscykler samt att skrivströmstyrningen är också låg efter 10^{10} programmeringscykler.

Tabell VIII
MINNESSPECIFIKATIONSKRAV FÖR SRAM-TYP APPLIKATION

Minnestyp	SRAM
Minnets chipstorlek	$>40\%$ mindre än SRAM
Uthållighet (antal programmeringscykler)	$>10^{10}$
Databevaringstid	$<1h$ (drifttemperatur på 85 - $150^\circ C$)
Dataminnestyp	Flyktigt minne
Läsoperationstid	$<50ns$
Skrivoperationstid	$<50ns$
Energi (pJ/bit)	$\leq 1pJ/bit$
Bitcellarkitektur	2T-2MTJ eller 1T-1MTJ

D. STT-MRAM minnesspecifikation

I Tabell IX visas minnesspecifikationen för minnestyp ett [37], minnestyp två [38] och minnestyp tre [3] av en STT-MRAM, som uppfyller minnesspecifikationskravet för en flash-typ och en SRAM-typ applikation. I [37] presenterades en 8 Mb vinkelrät STT-MRAM integrerad i en $28nm$ FDSOI, i [38] presenterades en 128 Mb vinkelrät STT-MRAM integrerad i en $40nm$ CMOS teknik respektive i [3] har en 1 Mb vinkelrät STT-MRAM integrerad i en $28nm$ FD-SOI presenterats, som en enda ersättande minnestyp för en mikrokontroller minnesarkitektur. För samtliga minnen i Tabell IX, framgår det att MTJ stackstrukturen har justeras med optimeringsprocessen och en peMTJ stack har använts för att uppnå förbättrade minnesegenskaper och samtidigt uppnå ett minskat slitage i bitcellerna för respektive STT-MRAM. Det framgår i [3] att en minnesarkitektur bestäms av CPU:n i en mikrokontroller. Det poängteras att en CPU med ett enda bussgränssnittssystem möjliggör att minnesarkitekturen bestående av en peSTT-MRAM minnestyp kan användas i en mikrokontroller, som både kodlagringsminne och som arbetsminne. Det innebär att i Tabell IX visas en minimal minneschiparea för minne tre på $58000\mu m^2$, som dessutom har en bitcellstorlek på $0.0364\mu m^2$ för en 1T-1MTJ bitcellarkitektur.

Minnescellarkitekturen som används i [37] [38] [3] för bitcellerna var en transistor seriekopplad med en peMTJ stackstruktur. Vidare uppges det att en maximal åstadkommen minnesuthållighet i [3] och en maximal åstadkommen databevaringstid i [37] på 10^{12} cykler respektive mer än 10 år som klarar drifttemperaturer upp till $125^\circ C$ kan uppnås. Det uppges att

den inbyggda STT-MRAM för minnestyp ett har ett maximalt resistent mot yttre magnetiska störningar upp mot 550 Oe under en minnesoperation, samt en TMR på $>150\%$ uppnås av samtliga STT-MRAM [37] [38] [3].

Tabell IX
MINNESSPECIFIKATION FÖR STT-MRAM

Minne	Minne 1	Minne 2	Minne 3
MTJ diameter		37nm	40nm
Bitcellarkitektur	1T-1MTJ	1T-1MTJ	1T-1MTJ
Bit bredd	64-bit		32-bit
Databevaringstid	$>10\text{år}$ (125°C)	$>10\text{år}$ (85°C)	$>10\text{år}$ (85°C)
Uthållighet (antal programmeringscykler)	$>10^6$ cykler (-40-125°C)	$>10^{10}$ cykler	$>10^{12}$ cykler
Densitet	8Mb	128Mb	128kB (1Mb)
CMOS teknik	28-nm FDSOI	40-nm CMOS	28-nm FDSOI
TMR	195%	$\geq 150\%$	$\geq 150\%$
Resistent mot magnetisk störning	550 Oe (85°C)		

E. STT-MRAM minnesprestanda

I Tabell X visas minnesprestandan för STT-MRAM, som en enda ersättande minnestyp i en mikrokontroller minnesarkitektur från [37] [38] [3]. För minne tre visas dessutom energiförbrukningen för en skrivoperation respektive en läsoperation per bit i en mikrokontroller. Det betyder att energikostnaden av en 128kB STT-MRAM för en 32-bit läsoperation och för en 32-bit skrivoperation är 29pJ respektive 96pJ. Tabell X visar att minne ett samt minne tre har en snabb läsoperationstid på $<5ns$ och endast minne två har en läsoperationstid på $<10ns$. Dessutom visar Tabell X en snabb skrivoperationstid på 10ns för minne tre, $\leq 14ns$ för minne två respektive $<20ns$ för minne ett.

V. DISKUSSION

A. Trovärdigheten av resultatet

Målet med studien var bland annat undersöka hur en eMRAM kan ersätta alla tre halvledarbaserade minnestyper i en mikrokontroller. I Tabell VII samt i Tabell VIII visas endast minnesspecifikationskravet för en flash-typ och en SRAM-typ applikation. Ett specifikt krav för en EEPROM-typ applikation kunde inte fastställas i studien. Dock framgår det i avsnitt II-E att ett flashminne är en utvecklad minnesvariant av EEPROM, vars minnescellstruktur är uppbyggd av FLOTOX, som är en variant av FGMOSFET. Dessutom framgår det i avsnitt II-A att båda minnen har som funktion att lagra instruktioner i en mikrokontroller och att de är lagringsminnen. Det som skiljer dessa två minnestyper är för ett flashminne sker snabba minnesoperationer blockvis och för en EEPROM sker det i byte. Då en eMRAM kan utföra snabba minnesoperationer

bitvis, innebär det att minnesspecifikationskravet för en flash-typ applikation kan tillämpas för en EEPROM-typ applikation och att målet kan anses vara uppfyllt.

Ytterligare en aspekt som kan påverka trovärdigheten i studien är om det presenterade data för energiförbrukningen av en STT-MRAM för endast minne tre i Tabell X är godtagbart. I Tabell V visas läs- och skrivenergiförbrukningen för en generell eMRAM från [9] och [30]. En jämförelse med läsenergin och skrivenergin i Tabell X visas det liknande låga energiförbrukningsnivåer som för minne tre. En bedömning av det presenterade data för energiförbrukningen kan därav anses vara trovärdig och godtagbar, om de angivna energiförbrukningarna från [9] [30] [3] anses bedömas vara korrekta. Slutsatser som presenteras i studien om eMRAM energieffektivitet kan därför anses vara lämpliga.

Tabell X
MINNESPRESTANDA FÖR STT-MRAM

Minne	Minne 1	Minne 2	Minne 3
Läsoperationstid	5ns	$<10ns$	5ns
Skrivoperationstid	$<20ns$	$\leq 14ns$	10ns
Läsenergi (pJ/bit)			0.9pJ/bit
Skrivenergi (pJ/bit)			3.0pJ/bit

B. Chiparea

En av frågeställningarna som skulle undersökas var hur lösningar kan skapas med mindre chiparea. I Tabell IX visas makrostorleken för minne tre från [3] av en 128kB STT-MRAM som har en chiparea på $58000\mu m^2$. En jämförelse med en samma kapacitet SRAM, visas i Tabell I att den har en chiparea på $204000\mu m^2$. Det betyder att en STT-MRAM makrostorlek står för nästan en tredje del av SRAM makrostorlek. Detta indikeras dessutom i Tabell V, som visar att en STT-MRAM fysisk cellstorlek är i de låga F^2 nivåerna och eftersom STT mekanismen är en skalbar teknik, betyder det att minnet kan skalas ner till ännu mindre cellstorlek och area i F^2 för tekniknoder lägre än 28nm. En begränsning med minnets skalbarhet var dock att en sådan egenskap inte var trivialt för att uppnå en hög E_b . Men det framgick att E_b förbättrades med en peMTJ och samtidigt kunde en minimal bitcellstorlek på $0.0364\mu m^2$ för en peMTJ STT-MRAM i [3] uppnås. Det betyder att en eMRAM som har en peMTJ stack kan skalas ner utan att det påverkar E_b egenskaper.

I jämförelse med de halvledarbaserade minnestyperna har en STT-MRAM en förenklad 1T-1MTJ bitcellstruktur. Det betyder att minnestypen kan uppnå liknande minnesegenskaper som SRAM eller som ett flashminne med en enda transistor. Dessutom har minnet möjligheten att ersätta hög densitet lagringsminnen som NAND-flash, tack vare att MTJ stacken är skalbart och att flera bitceller får plats i minneschipet. I jämförelse med de halvledarbaserade minnen vars begränsningar uppstår i bitcellstrukturen. Men även av att deras minnesegenskaper baseras till stor del på typen av transistor och dess funktionalitet. Det vill säga för dessa minnestyper är det därför en fråga om att optimera samt

utveckla transistorn och det innebär ytterligare komplexitet i bitcellernas arkitektur.

En lösning som minnet skapar med en signifikant mindre chiparea är att den framför allt kan användas som en enhetlig minneslösning för en mikrokontroller minnesarkitektur. Det medför en låg kostnad av den inbyggda minnets komponenter samt att det då skapar en minimal användandet av I/O operationer i ett system. Förutom att den totala kostnaden minskas i komponentkostnader blir dessutom den övergripande formfaktorn mindre tack vare minnesarkitekturen. Det beror dessutom på att flashminnet har två separata regioner för både kod- och datalagring som ger till följd att flashminnet kräver en större chip yta. Eftersom separata kod- och datalagringsregioner tillämpas för att tillåta att två minnesoperationer ska kunna utföras samtidigt. Det gör att kodlagringsregionen inte påverkas av att en datalagringsregion utför långsamma minnesoperationer. Det betyder att en STT-MRAM kan förenkla ett systems minnesarkitektur genom att minnestypen kan utföra minnesoperationer bitvis i nanosekunder. En eMRAM skapar därför även lösningar såsom en förenklad minnesarkitektur för icke-flyktiga minnesapplikationer, som möjliggör enhetlig minneslösningar för separata programkod- och datalagrings områden. Dessutom möjliggör minnet att en tillämpning av en sådan minnesapplikation inte behöver begränsas av chipstorleken.

C. Energieffektivitet

En annan frågeställning i projektet var att undersöka hur en eMRAM kan skapa lösningar med bättre energieffektivitet. I Tabell X visas det att minnestyp tre har en läsenergiförbrukning på 0.9pJ/bit samt en skrivenergiförbrukning på 3.0pJ/bit. Enligt specifikationskravet för en flash-typ applikation visar Tabell VII att energiförbrukningskravet ligger på ≤ 1 pJ/bit. Men kravet specificerar inte på om det är en läsenergiförbrukning eller en skrivenergiförbrukning, dock används ett flashminne för kodlagring samt för datalagring, dvs som ett läsminne. Det innebär att minnet används för att tillfälligt lagra och läsa instruktioner under en systemoperation i ett system. STT-MRAM uppfyller därav energiförbrukningskravet för en flash-typ applikation, som en enda ersättande minnestyp för ett systems minnesarkitektur.

En jämförelse mellan skrivenergiförbrukningen för en STT-MRAM och SRAM visas i Tabell I att SRAM minnet har en skrivenergiförbrukning på 0.73pJ/bit, vilket är mycket mindre än det som minne tre förbrukar. Enligt specifikationskravet i Tabell VIII uppfyller STT-MRAM därför inte kravet på ≤ 1 pJ/bit för en SRAM-typ applikation. Dock uppfyller minnet kravet på energiförbrukningen för en läsenergi per bit, som är mindre i jämförelse med SRAM läsenergiförbrukning på 0.96pJ/bit. Det innebär att minnestypens höga skrivenergiförbrukning kompenseras till viss del av den låga läsenergiförbrukningen. Därför är en SRAM-typ applikation för en STT-MRAM fortfarande lämplig i en mikrokontroller som ensamstående minnestyp, eftersom skillnaden i läsenergiförbrukningen mellan minnestyperna är en viktig faktor för inbyggda system som kräver frekventa läsoperationer.

En annan antagande som kan göras angående minnestypens höga skrivenergi är att det innebär en dubbelt så mycket skrivenergiförbrukning under en systemoperation och att minnet inte är lämplig som en SRAM-typ applikation i en mikrokontroller. Men detta bör inte gälla för en eMRAM, eftersom den har egenskaper som tillåter att minnet kan ersätta alla tre minnestyper i en mikrokontroller. Dessutom framgår det i avsnitt III att eMRAM inte har något strömläckage, som tillskillnad från SRAM har ständigt strömläckage på grund av SRAM flyktiga egenskaper. Det innebär att minnets låga skrivenergi bör inte antas som en enskild energikostnad och att strömläcket under drift bör tas med i den totala energikostnaden för SRAM minnet. Det innebär att den höga skrivenergiförbrukningen för en eMRAM inte har en stor betydelse som en förenklad enhetlig minneslösning för en mikrokontroller, eftersom den totala energiförbrukningen kommer att drastisk minskas i jämförelse med den generella minnesarkitekturen för systemet. Det innebär att extremt låga energiförbruknings mikrokontroller är fullt möjligt att utveckla samt att de existerande systemen kan optimeras. Det betyder att eMRAM även möjliggör en utveckling samt en optimering av inbyggda batteridrivna enheter som kräver långvarig självständighet. Det är en positiv konsekvens som framför allt beror på MTJ tekniken samt STT-mekanismen, som båda tillåter att minnestypen uppnår icke-flyktiga minnesegenskaper.

D. Läs- och skrivminnesoperationstid

En snabbare läs- och skrivoperationstid från eMRAM och hur en sådan minnesprestanda kan skapa lösningar var ytterligare en frågeställning som skulle undersökas. I Tabell X visas det att samtliga STT-MRAM har en läsoperationstid på < 10 ns respektive en skrivoperationstid på < 20 ns. En STT-MRAM uppfyller därav minnespecifikationskravet för en flash-typ och en SRAM-typ applikation. Om en jämförelse görs mellan SRAM och eMRAM, beskrivs det i avsnitt II-C att SRAM snabba minnesoperationstid är ett direkt resultat av att minnet har en BL och en \overline{BL} , som är konstant förladdade med en elektrisk laddning och det tillåter att den kan utföra minnesoperationer på nanosekunder. I avsnitt III-C framgår det att för en eMRAM är istället en hög TMR effekt en viktig faktor för att minnestypen ska uppnå en hög minnesoperationstid. Men en hög TMR effekt kan påverkas av hur frekvent en minnesoperation medför en misslyckande av en läsoperation. Då en sådan läsströmstyrning är anknuten till minnestypens STT mekanism och hur låg växlingsströmflöde som används under en minnesoperation. I avsnitt IV-A beskrivs det att en STT-MRAMs växlingsströmflöde kontrolleras genom att en peMTJ stack används och att en STT-MRAM minnesuthållighet är proportionell mot ett minskat växlingsströmflöde från STT mekanismen. Men samtliga STT-MRAM från Tabell IX har en hög minnesuthållighet på $> 10^{10}$ programmeringscykler, samtidigt som de har en hög TMR effekt på $> 150\%$. Det innebär att en STT-MRAM minnesoperationstid kan förbättras genom att ytterligare justera den peMTJ stacken samt att en hög TMR effekt inte påverkas av en hög minnesuthållighet.

Det indikerar på att en förenklad minnesarkitektur för en mikrokontroller inte bara är ett resultat av att STT-MRAM har

en mindre makrostorlek och därmed kräver mindre chiparea användning i ett system. Lösningen är också ett direkt resultat av att STT-MRAM har en läs- och skrivoperationstid som är lika snabb som SRAM, samtidigt som minnestypen har en obegränsad minnesuthållighet och uppnår minnespecifikationskravet för en SRAM-typ applikation samt för en flash-typ applikation. Det betyder att en enhetlig minneslösning är möjligt för en mikrokontroller samt att minnestypen då kommer att bidra för en låg total kostnad i komponentkostnader.

VI. SLUTSATS

En litteraturstudie som gjordes i projektet visade att en STT-MRAM uppfyllde minnesspecifikationskravet för en flash-typ och en SRAM-typ applikation. Lösningar som en STT-MRAM bidrar med är att minnestypen kan användas som en förenklad enhetlig minneslösning för en mikrokontroller minnesarkitektur. Det genom att STT-MRAM löser hårdvara och minnesprestanda begränsningar som kommer med separata kod- och datalagringsflashminnen, genom att STT-MRAM kan utföra snabba och bitvis minnesoperationer i nanosekunder. Dessutom medför lösningen en låg total energiförbrukning, genom att STT-MRAM ersätter det separata energiförbrukningen från respektive minnestyp, såsom SRAM, Flash och EEPROM för en mikrokontroller minnesarkitektur. Det betyder att en STT-MRAM möjliggör för extremt låga strömförsörjande batteridrivna enheter med långvarig självständighet. En snabb minnesoperation för minnet beror på STT-mekanismen samt den optimerade magnetiska lagringselement tekniken peMTJ. Det möjliggör att minnesuthålligheten och databevaringstiden kan kontrolleras samt justeras genom att en effektiviserad växlingsströmlöslösning uppnås, som är en viktig faktor för respektive minnesoperation. Det betyder att en STT-MRAM kan optimeras för ännu snabbare minnesoperations applikationer för en mikrokontroller.

TACK

Författaren vill tacka handledaren Gunnar Malm för bra vägledning under projektets gång.

REFERENCES

- [1] B. Krysiak. (2022, Apr.) The reason why there is a global semiconductor shortage. GLG, New York. [Online]. Available: <https://glginsights.com/articles/the-reason-why-there-is-a-global-semiconductor-shortage/>
- [2] (2022, Apr.) Embedded systems. Heavy.AI, San Francisco, CA, USA. [Online]. Available: <https://www.heavy.ai/technical-glossary/embedded-systems>
- [3] G. Patriceon, P. Benoit, L. Torres, S. Senni, G. Prenat, and G. Di Pendenza, "Design and evaluation of a 28-nm fd-soi stt-mram for ultra-low power microcontrollers," *IEEE Access*, vol. 7, pp. 58 085–58 093, May. 2019.
- [4] S. F. Barrett and D. J. Pack, *Microcontrollers Fundamentals for Engineers and Scientists*. San Rafael, CA: Morgan Claypool, 2006.
- [5] (2022, Apr.) Microcontroller memory types. Learning about Electronics. [Online]. Available: <http://www.learningaboutelectronics.com/Articles/Microcontroller-memory-types.php>
- [6] G. Malm, "Memories part 1," KTH, Stockholm, Sweden, 2021.
- [7] (2022, Apr.) Semiconductor memory. Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Semiconductor_memory
- [8] K. Zhang, *Embedded Memories for Nano-Scale VLSIs*, 1st ed. New York, NY, USA: Springer-Verlag, 2009.
- [9] S. K. Kurinec, *Nanoscale Semiconductor Memories : Technology and Applications*, 1st ed. Boca Raton, FL : CRC Press, 2017.

- [10] W. Sandqvist, "Halvledarminnen, mikrodatorn," KTH, Stockholm, Sweden, 2017.
- [11] (2022, Apr.) Samsung starts shipping 28nm embedded mram memory. MRAM-info. [Online]. Available: <https://www.mram-info.com/samsung-starts-shipping-28nm-embedded-mram-memory>
- [12] Z. Wang, X. Hao, P. Xu, L. Hu, D. Jung, W. Kim, K. Satoh, B. Yen, Z. Wei, L. Wang, J. Zhang, and Y. Huai, "Stt-mram for embedded memory applications," in *2020 IEEE International Memory Workshop (IMW)*, Jun. 2020, pp. 1–3.
- [13] A. Herland, "Källkritik med plagiering," KTH, Stockholm, Sweden, 2022.
- [14] D. Ary, L. Jacobs, C. Sorensen, and A. Razavieh, "Research approaches in education, reviewing the literature," in *Introduction to Research in Education*, 8th ed. Belmont, USA: Wadsworth, 2010, pp. 29–30.
- [15] (2022, Apr.) Basics of microcontrollers – history, structure and applications. Electronics Hub. [Online]. Available: <https://www.electronicshub.org/microcontrollers-basics-structure-applications/#Memory>
- [16] (2022, Apr.) Difference between volatile memory and non-volatile memory. GeekforGeeks. [Online]. Available: <https://www.geeksforgeeks.org/read-and-write-operations-in-memory/?ref=lbp>
- [17] (2022, Apr.) Endurance and data retention in non-volatile memories. doEEET. [Online]. Available: <https://www.doeet.com/content/eee-components/actives/endurance-and-data-retention-in-non-volatile-memories/>
- [18] S. Swanson, "Memories and sram," UCSanDiego, San Diego, CA, 2013.
- [19] S. Gangwar. (2022, Apr.) Read and write operations in memory. GeeksforGeeks. [Online]. Available: <https://www.geeksforgeeks.org/read-and-write-operations-in-memory/?ref=lbp>
- [20] (2022, Apr.) Static random-access memory. Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Static_random-access_memory
- [21] (2022, Apr.) Flash memory. Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Flash_memory
- [22] S. Muroga, "Ultra large-scale integration design," in *Encyclopedia of Physical Science and Technology*, 3rd ed., R. A. Meyers, Ed. New York, NY, USA: Academic Press, 2003, pp. 245–267. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B0122274105007961>
- [23] R. Micheloni and L. Crippa, "3 - multi-bit nand flash memories for ultra high density storage devices," in *Advances in Non-volatile Memory and Storage Technology*, Y. Nishi, Ed. Sawston, UK: Woodhead Publishing, 2014, pp. 75–119. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780857098030500035>
- [24] (2022, Apr.) Eeprom. Wikipedia. [Online]. Available: <https://en.wikipedia.org/wiki/EEPROM/>
- [25] L. Zhao, "Structural design of an electrically erasable eeprom memory cell," *World Journal of Engineering and Technology*, vol. 8, pp. 179–187, May. 2020.
- [26] (2022, Apr.) Conventional eeproms and flash eeproms offer a spectrum of bit densities. EDN. [Online]. Available: <https://www.edn.com/conventional-eeproms-and-flash-eeproms-offer-a-spectrum-of-bit-densities/>
- [27] (2022, Apr.) Highest-density eeprom memory device provides flexibility and reliability. Electropages. [Online]. Available: <https://www.electropages.com/2020/08/highest-density-eeprom-memory-device-provides-flexibility-and-reliability>
- [28] (2022, Apr.) Studying the program and erase cycle of an eeprom device. Comsol. [Online]. Available: <https://www.comsol.com/blogs/studying-the-program-and-erase-cycle-of-an-eeprom-device/>
- [29] T. Jew, "Mram in microcontroller and microprocessor product applications," in *2020 IEEE International Electron Devices Meeting (IEDM)*, Mar. 2020, pp. 11.1.1–11.1.4.
- [30] N. Maciel, E. Marques, L. Naviner, Y. Zhou, and H. Cai, "Magnetic tunnel junction applications," *Sensors (Basel, Switzerland)*, vol. 20, p. 121, Dec. 2019.
- [31] C. Tanaka, K. Abe, H. Noguchi, K. Nomura, K. Ikegami, and S. Fujita, "A scaling of cell area with perpendicular stt-mram cells as an embedded memory," in *2014 14th Annual Non-Volatile Memory Technology Symposium (NVMTS)*, Mar. 2014, pp. 1–3.
- [32] (2022, Apr.) Advanced mram technology – using mram in place of sram. Gsaglobal. [Online]. Available: <https://www.gsaglobal.org/forums/advanced-mram-technology-using-mram-in-place-of-sram/>
- [33] T. Y. Lee, K. Yamane, L. Y. Hau, R. Chao, N. L. Chung, V. B. Naik, K. Sivabalan, J. Kwon, J. H. Lim, W. P. Neo, K. Khua, N. Thiyagarajah, S. H. Jang, B. Behin-Aein, E. H. Toh, Y. Otani, D. Zeng, N. Balasankaran, L. C. Goh, T. Ling, J. Hwang, L. Zhang, R. Low, S. L. Tan, C. S. Seet, J. W. Ting, S. Ong, Y. S. You, S. T. Woo, E. Quek, and S. Y. Siah, "Magnetic immunity guideline for embedded mram reliability to realize mass production," in *2020 IEEE International Reliability Physics Symposium (IRPS)*, Jun. 2020, pp. 1–4.

- [34] E. Deng, Y. Wang, Z. Wang, J.-O. Klein, B. Dieny, G. Prenat, and W. Zhao, "Robust magnetic full-adder with voltage sensing 2t/2mtj cell," in *Proceedings of the 2015 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH '15)*, Aug. 2015, pp. 27–32.
- [35] X. Hu, D. Li, Y. Wang, J. Feng, Z. Ma, S. Wang, T. Min, X. Zeng, and Y. Xie, "An 8kb 40-nm 2t2mtj stt-mram design with 2.6ns access time and time-adjustable writing process," in *2021 IEEE 14th International Conference on ASIC (ASICON)*, Dec. 2021, pp. 1–4.
- [36] S. H. Han, J. H. Lee, K. S. Suh, K. T. Nam, D. E. Jeong, S. C. Oh, S. H. Hwang, Y. Ji, K. Lee, K. Lee, Y. J. Song, Y. G. Hong, and G. T. Jeong, "Reliability of stt-mram for various embedded applications," in *2021 IEEE International Reliability Physics Symposium (IRPS)*, Apr. 2021, pp. 1–5.
- [37] Y. K. Lee, Y. Song, J. Kim, S. Oh, B.-J. Bae, S. Lee, J. Lee, U. Pi, B. Seo, H. Jung, K. Lee, H. Shin, H. Jung, M. Pyo, A. Antonyan, D. Lee, S. Hwang, D. Jang, Y. Ji, S. Lee, J. Lim, K.-H. Koh, K. Hwang, H. Hong, K. Park, G. Jeong, J. S. Yoon, and E. Jung, "Embedded stt-mram in 28-nm fdsoi logic process for industrial mcu/iot application," in *2018 IEEE Symposium on VLSI Technology*, Feb. 2018, pp. 181–182.
- [38] H. Sato, H. Honjo, T. Watanabe, M. Niwa, H. Koike, S. Miura, T. Saito, H. Inoue, T. Nasuno, T. Tanigawa, Y. Noguchi, T. Yoshiduka, M. Yasuhira, S. Ikeda, S.-Y. Kang, T. Kubo, K. Yamashita, Y. Yagi, R. Tamura, and T. Endoh, "14ns write speed 128mb density embedded stt-mram with endurance $>10^{10}$ and 10yrs retention@85°C using novel low damage mtj integration process," in *2018 IEEE International Electron Devices Meeting (IEDM)*, Jan. 2018, pp. 27.2.1–27.2.4.

CONTEXT F

POWER SYSTEM CONTROL

POPULAR DESCRIPTION

KNOCKOUT FOR THE BLACKOUTS

Did you know there is a risk your power will shut down whenever you plug in the charger to your brand-new electric car? Has it ever occurred to you how difficult it is to supply the entire house with electricity? This becomes even more difficult when energy sources are becoming greener. By utilizing new technologies in the electricity grid, such blackouts can be avoided.

Most people realize the technology in cars must change for them to be charged with electricity instead of running on petrol or diesel. Equally important is transforming the electrical system when coal or nuclear power plants are replaced with renewables such as wind turbines and solar power. The current electrical system is designed for traditional power plants, such as coal and nuclear. They are very reliable and contribute to a stable power system by generating sufficient electricity regardless of weather conditions. When these outdated technologies are replaced with wind and solar power, some of this stability is lost. To combat this, new technologies need to be developed to avoid blackouts.

Solar power and wind turbines are seen around every corner today and represent the fight against climate change. The drawback of most renewable energy sources is the weather dependency resulting in inconsistent power production. By introducing energy reserves like batteries and spreading out the electricity production over large geographical areas, a stable electrical system can be built.

Thus, the next time the family's electrical vehicle is in dire need of some energy, think about the underlying stability the engineers supply to society's electrical grid.

SUMMARY OF PROJECT RESULTS

In response to the climate crisis, many countries have shifted their focus to electrification and increasing the share of renewable energy in their power grids. Renewable energy sources, such as wind or solar, differ from traditional power sources in some key aspects leading to new challenges within power system control. Some areas leading to potential problems include decentralization, increased variation, and loss of inertia in the power system. Inertia is the natural resistance to change in velocity in moving objects and serves as a backup in case of power outages with the help of spare energy stored in rotating masses in individual turbine generators. In traditional power sources, these rotating masses are directly connected to the electricity grid and provide a controllable power output while wind and solar are converter-based, which means they are connected to the grid by power electronics. These changes present new demands on power systems and their control methods. Maintaining stability in the grid is important to minimize the impact of faults and secure reliable access to electricity. The project groups working in this context aimed to examine how increased shares of renewable energy affect power system stability and design, as well as to investigate different methods to improve these aspects.

Project group F1 examined the importance of inertia and ways of maintaining frequency stability in a low-inertia system. As traditional energy sources using turbine generators are being replaced by renewable energy sources, such as wind and solar power, the inertia of the power system decreases. This makes the system more vulnerable to disturbances, thus compromising its stability. By investing in methods of primary frequency control, sudden changes in frequency may be avoided. The aim in this project is to improve frequency stability for a low-inertia system. To achieve this, the group made use of a simplified simulation model of the Nordic power system consisting of nuclear and hydropower. The model included the use of battery reserves to improve the recovery of the frequency levels in case of disturbances in a low-inertia system. By

analysing frequency deviation, the batteries were dimensioned such that the deviations were contained within acceptable levels.

For future research, one could explore different options instead of battery reserves. Furthermore, other renewable energy sources could be analysed in addition to nuclear and hydro generation.

Project group F2 compared the Small Signal Stability in a grid with high penetration of variable renewable energy (VRE) and a grid with no penetration of VRE and tested different stabilizing methods for the two cases. The stability of energy sources based on synchronous generators relies largely on the inherent amount of mechanical inertia. VRE sources are instead converter-based and do not have any inertia at all which lowers the system's stability. The group integrated two types of stabilizers into the grid to dampen the unstable oscillations. Power system stabilizers (PSS) were used to stabilize the grid with synchronous generators. To stabilize the grid with some penetration of VRE, PSS was used on the remaining synchronous generators. The group then analysed how stabilizers based on grid forming converters (GFM) perform. The project group found that controlling the grid with PSS was sufficient, both with and without VRE penetration. However, for the grid with a higher penetration of VRE, a much shorter settling time was possible utilizing a GFM.

In future research projects about this topic, the investigation of GFM-based stabilizers could be deepened further since the design of these is a promising albeit largely uncharted territory.

Project group F3 studied microgrid technology and chose to further analyse and simulate a DC microgrid since this seems to be the future version of microgrids. The simulation consisted of a photovoltaic (PV) system, a battery, a grid connection and residential loads. The project focus was on the power electronics and the controllers connecting the subsystems. The simulations showed one possible design of a microgrid for a house or building that uses a PV source and is also connected to the main grid. It was designed so it can work both in connection with the main grid as well as in isolated mode, disconnected from other energy supplies.

Future improvements of the project might be to add more energy sources to the microgrid, like a small wind turbine, or optimization of the Proportional Integral controllers used in the system control to increase stability and speed. To further improve the efficiency of the microgrid, a more complex coordinated control could be implemented. For example, a possible improvement is to include a weather forecast to foresee how much power the PV system will produce. Another possible project could be to verify the simulation results by implementing the system with hardware.

IMPACT ON SOCIETY AND ENVIRONMENT

To properly account for the effects of emerging technologies engineers must reflect on the global impact from both environmental and societal perspectives. Due to the need for increased integration of renewable energy sources, many sectors are going to be affected. Apart from the obvious improvements from these energy sources, they are also going to either directly or indirectly impact societies, industries, individuals, and environments both positively and negatively.

Improved control methods will allow higher shares of renewable energy production in the power system. This will contribute to reducing greenhouse gas emissions from the energy sector and thereby contribute to mitigating the effects of climate change. This is also beneficial to communities whose living conditions are negatively affected by emissions. Incorporating more renewable energy production is also important for increasing the capacity of the grid. This will be important in the future to further decrease greenhouse gas emissions due to consumers using more power as many technologies are being electrified.

There are also some concerns with the implementation of renewable energy which are important to consider. For example, the ecological effects of implementing wind power, such as the effect on habitats for birds and bats, can have grave consequences for endangered species. The competition of land use with other sectors can also be problematic, for example between solar parks and agriculture.

Almost everyone agrees that we need to implement more renewable energy sources to reduce greenhouse gas emissions and wind power can contribute to that. However, residents usually complain about the appearance and noise level of such power plants near residential areas even if they produce renewable energy. The placement of these power plants can furthermore affect both liveability and the housing market in those areas. On the other hand, if all citizens are accommodated, then the ecology and wildlife might suffer instead. By considering these effects when planning the placements of renewable energy power plants these negative effects can be minimized but not eliminated.

Another aspect of increased integration of renewable energy sources is the sheer number of raw materials needed to construct all parts in a stable system. For example, when manufacturing batteries, which are used both in micro and macro systems, a lot of rare elements are needed. The excavations required are detrimental to the environment since the earth's natural resources are being exhausted, but also because the land nearby is both destroyed by the digging and may also be poisoned by chemicals polluting the lands and waters. The mining industry will probably flourish as the shortage of some of them will make them more profitable to extract. Of course, this can affect health and the living conditions negatively for those living nearby, the miners or have their income from the expropriated land. For the environment and sustainability, it will be more important to recycle the materials when the facilities are scrapped.

Individuals can benefit from investing in microgrids both financially and from improved stability, in case of main grid faults. Photovoltaic power sources are already a proven technology with reasonable payback times, far shorter than the expected lifespan of the system. On the other hand, the initial investment is often large and requires capital or funds from external stakeholders. The benefits include distributed production of renewable energy as well as a source of power for the individual owners even when the main grid experiences blackouts. This leads to both improved safety and convenience for individuals.

When implementing renewable energy sources, the control systems will likely turn more digital. On one hand, this will provide the power grid with a faster and more efficient transition between power plants. On the other hand, a digital approach requires extensive cyber security to safely provide citizens with electricity in critical sectors such as hospitals. However, an increase in cyber security along with an expanded renewable energy grid will also provide additional employment opportunities. On the other hand, the transition from outdated energy sources would also abolish older jobs.

The implementation of renewable energy sources such as wind, solar and hydro generation are important aspects to combat climate change. However, these implementations include difficulties on various levels. To make this change in the electrical power system with as few negative consequences as possible it is important to consider all parties. Neither individuals, society nor the environment should face too many sacrifices.

Supporting Frequency Stability With Batteries in Low Inertia Power Systems

Tim Asking and Sophie-Linn Karlsson

Abstract—As the share of power electronics-based renewable energy sources increases in power systems, the system inertia provided by conventional generation is reduced. Inertia is an important factor in the grid's frequency stability, and with its reduction comes challenges to ensure the reliability of the grid. The frequency stabilising service of frequency containment reserves will need to work in conjunction with the faster, stabilising service of fast frequency reserves to avoid power failures in case of sudden disturbances. This project aims to examine the impact of inertia and methods of improving frequency stability in a future low inertia power system. The frequency behaviour is studied using a simplified and linearised model of the Nordic power system implemented in Matlab/Simulink. The model is extended by implementing supplementary battery control to support the frequency response. The simulation results show that there is an evident correlation between the reduction of system inertia and frequency instability. Moreover, it is concluded that the implemented battery support is successful in stabilising frequency following a disturbance.

Sammanfattning—Då andelen kraftelektronikbaserade förnybara energikällor ökar i kraftsystem så kommer systemets tröghet, tillfört av konventionell generering av elektricitet, att minska. Trögheten är en viktig faktor för elnätets frekvensstabilitet. Då trögheten minskar så utmanas tillförlitligheten av elnätet. Frekvensstabiliserande frekvenshållningsreserver behöver fungera i samspel med de snabbare och stabiliserade frekvensreserverna för att undvika strömvabrott vid plötsliga störningar. Projektet ämnar undersöka trögheten och metoder som används vid förbättring av frekvensstabilitet i framtida kraftsystem med låg tröghet. Beteendet hos frekvenser studeras med en förenklad och linjäriserad modell av det nordiska kraftsystemet implementerat i Matlab/Simulink. Modellen utökas genom att inkludera en batterikontrollmetod för att tillföra ett snabbt frekvenssvar. Resultatet av simuleringarna visar att det finns en korrelation mellan minskning av systemets tröghet och frekvensinstabilitet. Vidare visas det att implementationen av batteristöd lyckas förbättra frekvensen i fallet av en störning.

Index Terms—Battery Power Support, Fast Frequency Reserves, Frequency Containment Reserves, Inertia, Renewable Energy Sources

Supervisors: Danilo Obradović

TRITA number: TRITA-EECS-EX-2022:140

I. INTRODUCTION

A. Background

As the consequences of climate change become evident, efforts are made to develop more sustainable ways of living. Striving towards a sustainable society has brought changes to the energy sector. In the Nordic electricity grid, there is a mixture of Renewable Energy Sources (RES) and conventional

energy sources, including nuclear, hydro, and thermal power [1]. Being driven by climate policies, the Nordic power system is expected to further expand wind power and close down thermal power plants in the future [2]. This increase in demand for power electronics-based RES, such as wind and solar power, requires the Transmission System Operators (TSOs) to adapt their power systems to stay fully operational [2].

Conventional means of energy production such as thermal, hydro, and nuclear power plants use generators that are synchronously connected to the power grid. These synchronous generators contribute to system inertia. System inertia refers to the amount of kinetic energy stored in the rotating masses that are connected to the system [3]. Whenever there is an imbalance between the amount of generated active power and consumed active power, the system frequency deviates from the nominal value of 50 Hz. A power system with sufficiently large inertia can resist sudden changes in frequency by absorbing or injecting kinetic energy into the system [4]. On the other hand, power electronics-based RES are connected to the grid via power electronics. This decouples their generators from the grid and makes them operate independently of the system frequency [3]. As a result of their decoupling, power electronics-based RES do not contribute to system inertia naturally. Therefore, in the presence of their high penetration, careful control tuning of implemented frequency controllers is needed.

Due to the varying nature of the generation and consumption of electricity, frequency control needs to be implemented to keep the system frequency within the acceptable interval. One such frequency control method is Frequency Containment Reserves (FCR). FCR is an operational action that increases generation when the frequency drops below a defined value. By using FCR, the frequency can be contained within acceptable levels of Steady-State Frequency Deviation (SSFD) and maximum Instantaneous Frequency Deviations (IFD) [4]. Together with the Rate of Change of Frequency (RoCoF), IFD and SSFD are key parameters when studying frequency behaviour. The Nordic power system is designed to fulfil the N-1 criteria, where a system of N components should remain operational even if one component fails, leaving N-1 components in the system [5]. The system aims to quickly restore the operation after a disturbance so that it can handle any new disturbances that severely disrupt the power balance and produce large frequency deviations [5]. Large frequency deviations are not tolerated by the system. In [4], it is mentioned that this may activate protective systems, causing system separation, loss of load and production units, as well as customer outages. FCR is one frequency control method

targeted on large disturbances, and it may be coupled with one or several different control methods to restore the balance between powers in the case of disturbances.

One method of supporting grid stability is by using supplementary power supplies. Batteries can be used to store power which can be delivered to the system at the time of a deficit in generated power. The fast response time of batteries enables them to react to a frequency deviation in a short amount of time [6]. This property makes batteries a suitable source of Fast Frequency Reserves (FFR). FFR is similarly to FCR a service aimed to stabilise frequency, but it is activated several times faster [6]. FFR is a complement to FCR, and the first activated service when a disturbance occurs in a low inertia system [7]. In a system with varying frequencies, FFR is important to quickly restore the balance between generated and consumed power and to avoid power outages.

Research on how batteries can be used to efficiently support stability in the grid is ongoing. FFR control can be implemented in various ways. In [8], the effects of implementing a fast power reserve in a low inertia system using batteries were analysed. Another example is to use droop control with power converters to simulate the behaviour of a synchronous generator [9]. It is also possible to provide synthetic inertia through grid-connected power converters [10]. In this report, FFR using batteries is implemented in a low inertia system with parameters tuned to correspond to the Nordic power system.

B. Project Set Up and Goals

The project examines the effect of inertia using a linearised model in Matlab/Simulink of the Nordic power system. The model consists of two hydropower units and two nuclear power units which are run under different simulation conditions, also referred to as cases. Each simulation case uses a different amount of system inertia.

Firstly, the concept of Frequency Containment Reserves for Disturbances (FCR-D) is presented by explaining the fundamental behaviour of mechanical and load power after a step disturbance. The changes in mechanical and load power are further examined through simulations to explain deviations in the frequency response. Similar simulations are run for the system with implemented battery support. The results are analysed and compared to a second control method, where certain system parameters are tuned to fulfil the frequency requirements for the case of low system inertia. In [4], it is mentioned that the maximum IFD should not deviate more than 0.9 Hz, and SSFD not more than 0.4 Hz, from the nominal frequency of 49.9 Hz for when FCR-D is activated. Lastly, a discussion on some practical aspects is presented. These aspects include an analysis of the system viability, a simple model used to calculate battery cost, a comparison to other solutions for improving frequency stability, and some suggestions for future research.

II. FREQUENCY CONTAINMENT RESERVES

A. FCR in the Nordic TSOs

The Nordic TSOs consist of Denmark's "Energinet", Finland's "Fingrid", Norway's "Statnett", and Sweden's "Svenska

Kraftnät", all sharing a high voltage grid [11]. Apart from the Nordic TSOs, each country also has high voltage grid connections to other operators outside the Nordics. Furthermore, in [12] it is mentioned that the Nordics are part of the European Network of Transmission System Operators for Electricity (ENTSO-E) and henceforth complies with their operational limits for frequency levels [13].

All FCR providing units need to follow some performance requirements, for increased reliability and security, to participate in the FCR market. In [14], a validation process is described where the FCR units are required to stay within nominal frequency levels for different processes, such as normal operation and for system disturbances. The frequency is allowed to deviate slightly, but the allowed magnitude of the deviation depends on whether the units run in normal operation or if they are reacting to any kind of disturbances. The Nordic TSOs mainly use FCR in hydropower plants given that approximately half of the power production consists of hydropower [15]. The benefit of FCR from hydropower comes from its reliability to produce power in most conditions. The production of electricity is regulated using governor control. Implementing the necessary governor control for FCR in nuclear power plants is not preferable, since limiting the power output of a nuclear power plant may cause operational issues [7].

In [14], two products are defined for the FCR market. One of them is Frequency Containment Reserves for Normal operation (FCR-N) and the other is FCR-D. FCR-N both up- and down-regulates the frequency to stay within 0.1 Hz from its nominal value while FCR-D up- or down-regulates the frequency to stay within 0.5 Hz from its nominal value. However, FCR-D for upwards regulation is more commonly used given that under-frequency occurs more often.

In [14], it is mentioned that FCR-D will activate whenever the frequency deviates more than 0.1 Hz from the nominal frequency and remain active for the full duration of the disturbance. Additionally, FFR can be activated when the frequency drops with 0.3 Hz to 0.5 Hz from the nominal frequency [16]. The minimum frequency should not drop below 49 Hz whereas a frequency lesser than 47.5 Hz will automatically disconnect power units to avoid damage [4]. The frequency is a measure of the power balance between generated and consumed active power in the system, and it should be close to its nominal value to ensure the safe operation of the system.

B. Inertia and Frequency Stability

Inertia, which was briefly mentioned in Section I, is an important property of frequency stability. Rotational inertia is stored in synchronous generators connected to the power grid, and it is used to oppose sudden changes in frequency. By releasing or absorbing energy, the synchronous generators provide a response to power imbalances [17]. When the power demand exceeds the power production, there will be a deficit in power. Synchronous generators respond to this power deficit by releasing kinetic energy stored in their rotating masses to the grid. The loss of kinetic energy will lead to a temporary decrease in frequency. The rate at which frequency

decreases depends on the size of the imbalance in power and on the system inertia, where high system inertia decreases the rate of change [3]. With the conversion from energy generation using synchronous generators to RES with power electronics, alternative solutions, such as FFR, are needed to compensate for the loss of inertia. An alternative is to use synthetic inertia, which simulates the inertia's behaviour through methods such as Battery Energy Storage System (BESS), High-Voltage Direct-Current grid (HVDC) that links outside the synchronous area, or modulation of the power output of inverters used in wind turbines [2]. However, the method used for this project will be FFR for the given system model.

During the Inertial Frequency Response (IFR), the first drop in frequency after a disturbance, both RoCoF and IFD are studied. After a large disturbance in the system, both RoCoF and IFD are of importance since an increase of these properties could cause load-shedding [4]. Maximum RoCoF is the tangent to the steepest time derivative of the frequency during IFR and is defined as:

$$\frac{\partial f(t_d)}{\partial t} = \frac{\Delta f(t_d)}{\Delta t} \quad [\text{Hz/s}] \quad (1)$$

where $t = t_d$ is the time when a disturbance occurs. Maximum IFD on the other hand, is measured as the maximum frequency deviation during IFR and is defined as:

$$\Delta f = f_{\min} - f_s \quad [\text{Hz}] \quad (2)$$

where f_{\min} is the lowest frequency during the activation of FCR. Furthermore, RoCoF and IFD are parameters that are heavily affected by the inertia levels as can be seen in Fig. 1, where high and low system inertia is compared in case of a step disturbance in the system.

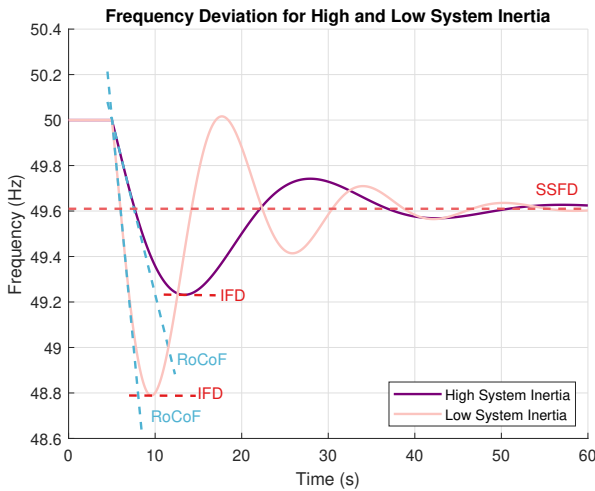


Fig. 1: Frequency response after a step disturbance, with FCR-D control, for high and low system inertia.

Moreover, Fig. 1 shows that high system inertia ensures lower frequency deviations compared to low system inertia where the deviations are greater. For lower levels of inertia, both RoCoF and maximum IFD will increase, while they will

decrease for higher levels of inertia. An increased value of RoCoF and IFD will increase the likelihood of the system becoming unstable. It can also be noted that the SSFD remains the same for both cases.

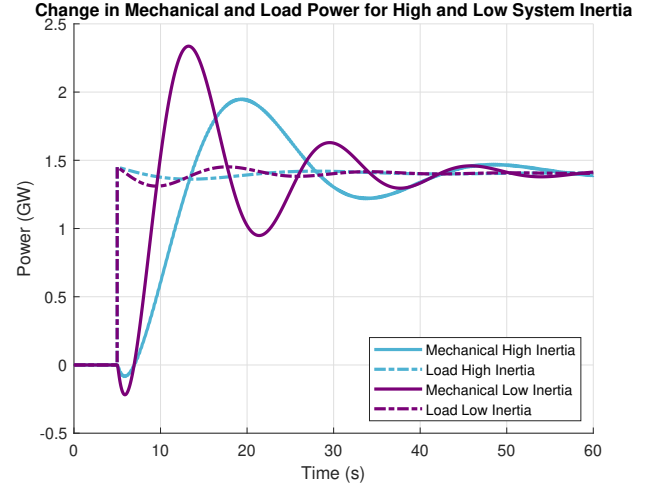


Fig. 2: Change in mechanical and load power after a step disturbance, with FCR-D control, for high and low system inertia.

In Fig. 2, the change in mechanical and load power for the system is shown for the same simulation case seen in Fig. 1. By comparing Fig. 1 and Fig. 2, it is seen that the frequency drops at the same time as when the load power exceeds the mechanical power. It can also be seen that the frequency increases from the maximum IFD at the same time as when the mechanical power exceeds the load power. The frequency reaches steady-state when there is a balance between the powers. Moreover, the amount of inertia is seen to have an effect on both mechanical and load power. For the low inertia case, the mechanical and load power has larger oscillations. It is also seen that the mechanical power increases faster in the low inertia case.

C. Impact of Parameters on FCR-D

Apart from the impact of inertia on RoCoF, maximum IFD, and SSFD, a few other parameters are studied to see how they affect the frequency response of the system. These parameters include the damping constant D , largest power disturbance ΔP_D in the power system, water time constant T_W for the hydropower as well as the system gain R . These can be seen in Fig. 3. The figure only shows the case for low system inertia given that no major differences were observed compared to the case of high system inertia.

When looking at the effect of the damping constant on the system, it can be seen that both maximum IFD and SSFD decrease when increasing the damping constant, while RoCoF remains essentially at the same level. This seems reasonable given that damping should decrease deviation when regulating the system. For the power disturbance, simulations show that an increase in disturbance gives increased values for maximum IFD, SSFD and RoCoF. Increasing the water time constant provides an increase in maximum IFD and

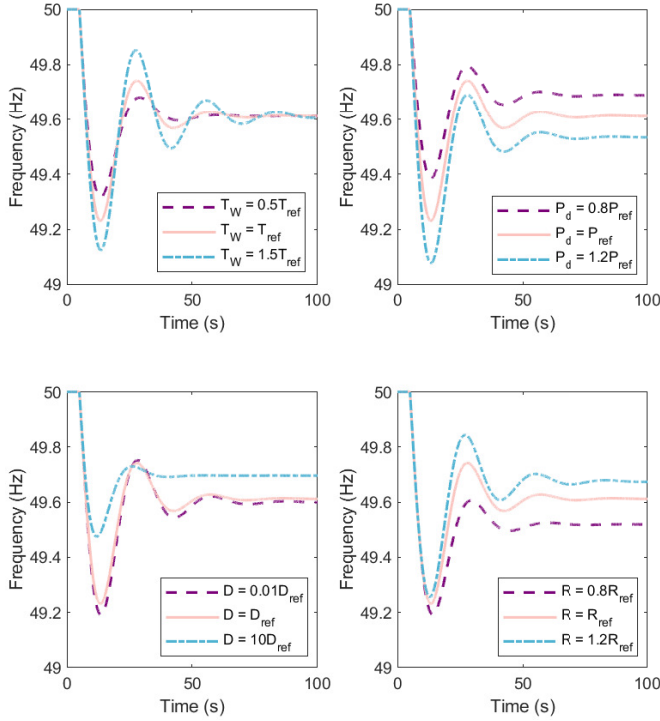


Fig. 3: Frequency responses as the parameters D , ΔP_D , T_W and R change for low system inertia.

increased oscillations. Nevertheless, both RoCoF and SSFD are seemingly unaffected by any changes to the water time constant. Since the water time constant exclusively affects the water turbines, increasing the constant gives slower turbines. Accordingly, it takes a longer time to react to deviations. Lastly, increasing the system gain R provides a decrease of both maximum IFD and SSFD, while RoCoF visually looks the same.

III. SYSTEM MODEL

A. The Swing Equation

To study the relationship between the mechanical and electrical power in a system containing synchronous generators, the swing equation is used. It describes the motion of the rotor in generator k , such as [4]:

$$\dot{\omega}_k = \frac{1}{M_k} (P_{mk} - P_{ek}) \quad (3)$$

where ω_k is the difference in rotor speed $\omega_k = \omega_{kg} - \omega_s$, with ω_{kg} referring to the electrical speed and $\omega_s = 2\pi f_s$ being the synchronous speed. f_s is the nominal frequency of 50 Hz. P_{mk} and P_{ek} are the mechanical respectively electrical powers of generator k . From the equation, it is evident that an imbalance between mechanical power and electrical power leads to a change in rotor speed ω , thus affecting the system frequency. Moreover, (3) verifies that the rate at which frequency changes is dependent on the size of the imbalance as well as the amount of inertia. All variables in (3) are in per units. Furthermore, M_k is defined as:

$$M_k = 2H_k \frac{S_{ngk}}{S_{base}} \quad (4)$$

TABLE I
INERTIA CONSTANTS M_i ($i = 1, 2, 3, 4$) (s) OF THE FOUR GENERATORS

M_1	M_2	M_3	M_4
9.70	7.70	11.70	13.70

where H_k is the inertia constant of generator k in seconds (s), S_{ngk} is the generator rated power in volt-ampere (VA), and S_{base} is an arbitrary base power. After a disturbance has occurred, the frequency may vary in different parts of the power grid [18]. To account for this effect and obtain an average representation of frequency for the system, the Center of Inertia (COI) reference frame can be used [18]. Equation (3) is transformed to its COI equivalent and adapted to a system of n_g generators, giving the equation:

$$\dot{\omega}_{COI} = \frac{1}{M_T} \sum_{k=1}^{n_g} (P_{mk} - P_{ek}). \quad (5)$$

In (5), M_T corresponds to the system inertia and equals the sum of M_k for the individual generators in the system. The model considers small deviations from the initial values, represented by Δ . Any damping in the system may be included in (5) by introducing the damping constant D_{COI} . By letting $\dot{\omega}_{COI} = \dot{\omega}$, $M_T = M$, and $D_{COI} = D$, the following equation is obtained:

$$M\Delta\dot{\omega} = \Delta P_m - \Delta P_L - D\Delta\omega. \quad (6)$$

In (6), ΔP_L is the non-frequency-sensitive load change while $D\Delta\omega$ is the frequency-sensitive load change, including the damping constant D [4]. The change in electrical power, also referred to as load power, can be expressed in terms of the load change with the following equation:

$$\Delta P_e = \Delta P_L + D\Delta\omega \quad (7)$$

B. Modelling

The project uses a linearised model with parameters tuned to correspond to the FCR-D dynamics of the Nordic power system. The Simulink model used to run simulations includes two equivalents of hydropower plants, HP1 and HP2, as well as two equivalents of nuclear power plants, NP1 and NP2. The generators have different inertia constants M_i , as seen by Table I. FCR-D is included in the hydropower plants, which uses a feedback loop to enable governor control.

Four different inertia cases are simulated. In inertia case 1, all power units HP1, HP2, NP1, and NP2 are active. In inertia case 2, all but NP1 are active, and in inertia case 3, all but NP2 are active. Moreover, in inertia case 4, neither NP1 nor NP2 are active. By deactivating power units, the system inertia is affected. The most critical inertia case is case 4, where the system inertia is reduced the most. Deactivating the nuclear power plants could be seen as replacing the generators with power electronics-based RES, as they do not contribute to the system inertia. A step disturbance is simulated by suddenly increasing the load to a maximum of 1.45 GW.

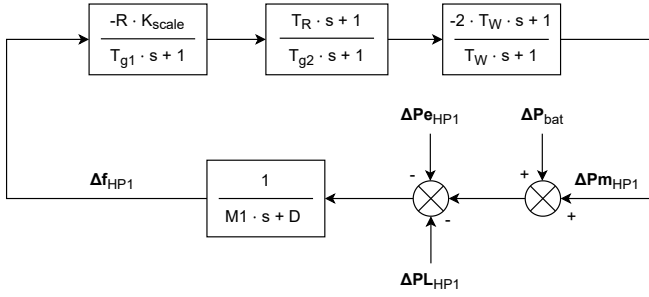


Fig. 4: Implementation of the linearised hydropower unit HP1 in Simulink.

Fig. 4 shows the implementation of the hydropower unit HP1. The model includes the transfer functions representing the dynamics of the hydro turbine, the governing system used for FCR-D, and the swing dynamics. In [4], the transfer function representing the dynamics of the hydro turbine is derived using Bernoulli's equation. $\Delta P_{m_{HP1}}$ and $\Delta P_{e_{HP1}}$ represent the change in mechanical and electrical power of HP1 respectively. The step disturbance $\Delta P_{L_{HP1}}$ creates a sudden increase in load power that causes significant frequency deviations. ΔP_{bat} represents the supplementary battery power. Δf_{HP1} corresponds to the change in frequency. The model uses the following parameters:

- R , representing the system gain, the inverse of the speed droop g_{st}
- T_W , representing the turbine effective water constant
- D , representing the damping constant originating from the load
- g_{tr} , representing the gain of the transient feedback loop
- T_R , representing the time constant of the transient feedback loop
- T_G , representing the time constant of the control servo

Moreover, the parameters T_{g1} and T_{g2} are calculated as:

$$T_{g1} = \frac{K_p + R + \sqrt{(K_p + R)^2 - 4T_G R}}{2K_i}$$

$$T_{g2} = \frac{K_p + R - \sqrt{(K_p + R)^2 - 4T_G R}}{2K_i}$$

The model also includes the parameters K_p , K_i , and K_{scale} , all related to the governor control. The usage of these parameters will be further discussed in Section V.

Fig. 5 shows the implementation of the nuclear power unit NP1. The model only includes a transfer function representing the swing dynamics by considering the inertia constant M_3 .

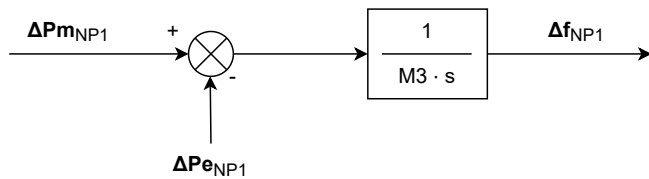


Fig. 5: Implementation of the linearised nuclear power generator NP1 in Simulink.

IV. BATTERY SUPPORT

A. Previous Research

The possibility of using batteries to stabilise frequency has been evaluated in several reports. In [19], the provision of primary frequency response from Electric Vehicles (EV) is examined for a future low inertia system. Appropriately used charging strategies are shown to improve frequency stability by decreasing the EV charging load at the time of a generation outage. The report also discusses the possible economic and environmental benefits of frequency support from EVs. Similarly to this project, [20] uses a battery control model to investigate the frequency support provided on a grid level transmission system by injecting active power. It is shown that BESS can improve the frequency support, but that the injected power may be negatively affected by a transmission system's high impedance. The placement of the BESS in the transmission system is thus of importance. Moreover, the controller gain needs to be suitably adjusted to avoid a sudden discharge of the battery. A high controller gain may lead to a frequency above the nominal value, which means that the methods of frequency control used must cooperate to stabilise the grid frequency within the allowed limits.

B. Simulation

To investigate the effect of implementing FFR using batteries, a Simulink model of a proposed battery control method is used. The battery control method is adapted from [7]. In the Nordic power system, FFR can take the form of an increase in the injected power or as a load reduction when the frequency is below the nominal value [16]. Support from FFR can have different duration, where a long support duration is activated for at least 30 s while a short support duration is activated for at least 5 s [16]. In this project, the short support duration is examined. Providers of FFR in the Nordic power system need to provide their services following predefined requirements of frequency activation level and maximum full activation time.

In this project, FFR is implemented as an additional injection of active power. The increase in power results from the activation of an external battery of a certain battery power capacity P_b . The curve of the battery power is made to have a trapezoidal shape with a maximum full activation time t_{fa} , a support duration $t_{support}$ and deactivation time t_{deact} of 7.5 s, and a battery converter time constant T_c of 0.05 s. The battery support is activated whenever the frequency surpasses the frequency threshold $f_{threshold}$. The block diagram of the battery control method under study is shown in Fig. 6.

In Fig. 6, the trigger signal is a logical signal that reaches the value 1 at the time of the battery activating. It is dependent on the defined values of $f_{threshold}$ and $f_{nominal}$, as well as on the frequency deviation Δf . For simulation purposes, a variable *state* is included to indicate whether or not the battery support is included in the system model. The delay block delays the trigger signal by the sum $t_{fa} + t_{support}$. The blocks "Slope and power up" and "Slope and power down" use ramp signals with adapted slopes that, once combined, produce a trapezoidal shape. The signal is then passed through a block representing the dynamics of the battery using a first-order

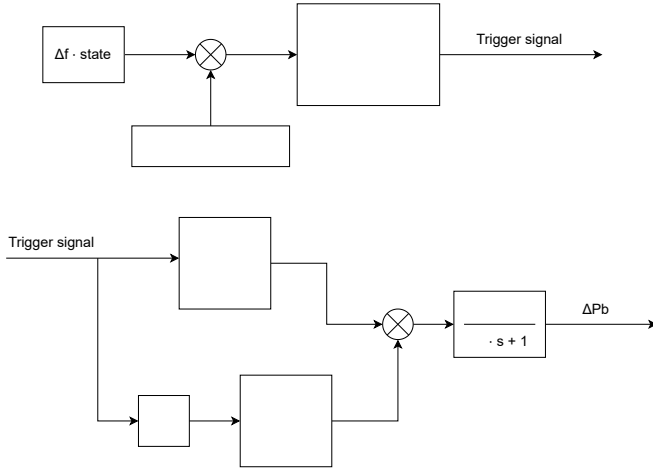


Fig. 6: Block diagram showing battery controller implemented in Simulink

transfer function. The resulting battery power ΔP_b is shown in Fig. 7.

For each of the three activation cases, proposed in [7], P_b was adapted to give a maximum IFD of no more than -0.9 Hz for the low inertia case 4 and a disturbance of 1.45 GW. The final parameters used for the simulations that follow are shown in Table II.

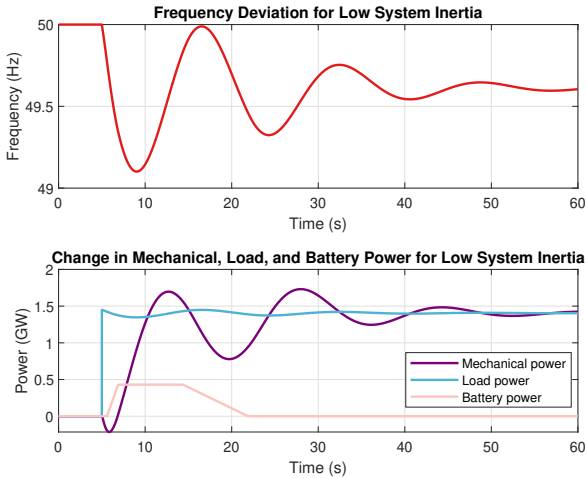


Fig. 7: Frequency response and change in power, including battery power, for the low inertia case 4.

TABLE II
PARAMETERS USED IN BATTERY SIMULATIONS

Activation case	$f_{threshold}$ (Hz)	t_{fa} (s)	P_b (GW)
1	49.7	1.3	0.43
2	49.6	1.0	0.44
3	49.5	0.7	0.45

The proposed activation cases in Table II were examined by comparing minimum frequency, maximum provided power from the battery, and the energy from the battery. Simulations were made for 17 different conditions, where the power disturbance and inertia case varied. A bar chart showing the

minimum frequency for the activation cases under the different simulation conditions is shown in Fig. 8.

As seen in Fig. 8, activation case 1 gives a higher minimum frequency due to its activation threshold. It is also seen that the values of $f_{threshold}$ may cause the battery to not activate, as evident by the last two measurements. In the case with 250 MW disturbance and inertia case 1, the frequency does not drop below any of the three frequency thresholds and can be stabilised using FCR only. In the case with 250 MW disturbance and inertia case 4, the battery is activated for activation case 1. However, the minimum frequency is lower in activation case 1 than for cases 2 and 3. This is because the increase in power that the battery delivers results in a surplus of generated power, thus leading to a frequency higher than the nominal value. The simulations also show that the maximum provided power in all activation cases corresponds to P_b . Moreover, as the maximum provided power is independent of the simulation conditions, the battery energy is constant for all activation cases. The battery energy is 5.12 GWs, 5.17 GWs, and 5.22 GWs for activation cases 1, 2, and 3 respectively.

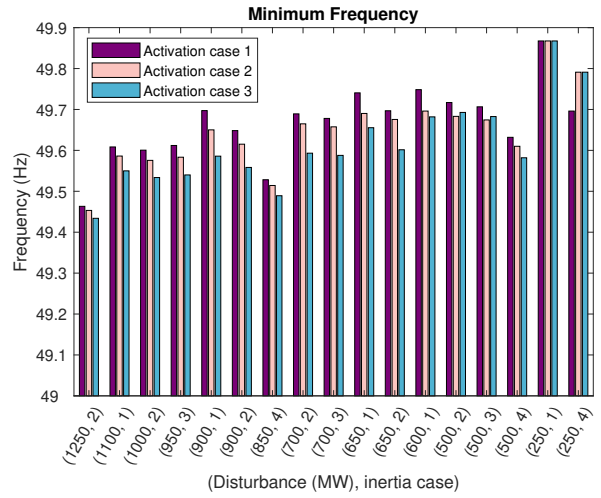


Fig. 8: Bar chart showing the minimum frequency for three activation cases and different simulation conditions.

V. FREQUENCY CONTROL WITHOUT BATTERY SUPPORT

A. K -parameters

In addition to the simulations for FFR using battery support, brief simulations to adjust parameters K_p , K_i , and K_{scale} are made to analyse their viability as an alternative control method. The K -parameters are part of the hydropower control loop seen in Fig. 4. Hence, it is possible to control FCR-D by adjusting them. K_{scale} is used directly in the control loop while the K_p and K_i parameters are part of T_R :

$$T_R = \frac{K_p}{K_i} \quad (8)$$

from which T_R is a time constant used for a part of the governor control. The benefit of adjusting the frequency level directly in the control loop compared to battery support lies in the decreased cost of such a control method.

B. Simulation of Alternate Control Method

The goal is to achieve a maximum IFD no greater than -0.9 Hz. The initial values of the K-parameters for the system model are:

$$K_p = 4.97, K_i = 2.61, K_{scale} = 0.47 \quad (9)$$

In Fig. 9, Fig. 10, and Fig. 11, the reference values equals the nominal K-parameters seen in (9).

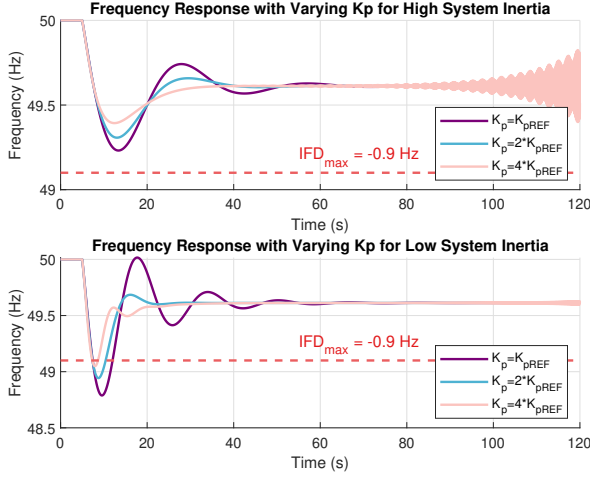


Fig. 9: Frequency response for different levels of K_p where K_{pREF} is the nominal value used in the system model.

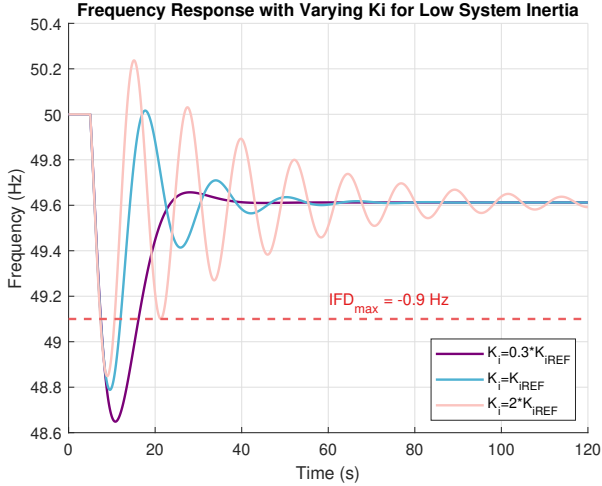


Fig. 10: Frequency response for different levels of K_i where K_{iREF} is the nominal value used in the system model.

In Fig. 9, the limit for maximum IFD, $IFD_{max} = -0.9$ Hz, will be exceeded for any value of K_p for low system inertia. To decrease the maximum IFD, K_p must be increased from its nominal value. For high system inertia, the deviation will not breach the limit for maximum IFD. However, the system will start oscillating for larger values of K_p . Hence, neither a sufficient IFD nor a stable system can be achieved at the same time.

In Fig. 10 and Fig. 11, both parameters exceed the limit for maximum IFD in a low inertia system for all respective

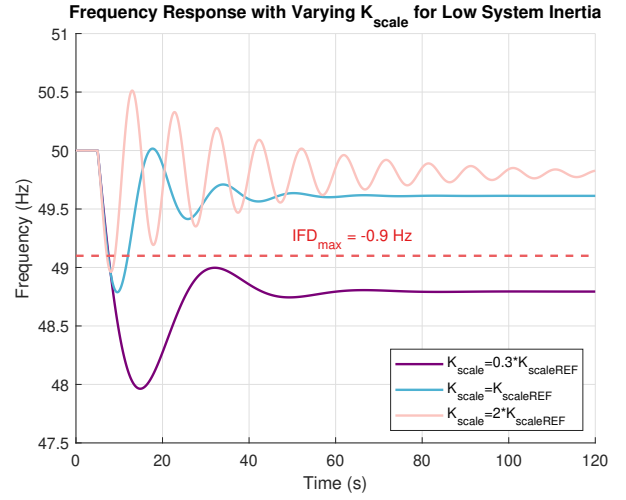


Fig. 11: Frequency response for different levels of K_{scale} where $K_{scaleREF}$ is the nominal value used in the system model.

values. Additionally, given that an increased K_i and K_{scale} prolongs the settling time, while still exceeding the maximum IFD limit, they are also deemed unfit to regulate the system.

In [4], it is noted that T_R should be a few seconds. For the system model, $T_R = 1.903$ s is within its limitations. For the K_p and K_i parameters' values, it can be noted that T_R will increase past its limit when K_p is increased. This goes well with the results above where it was noted that K_p should increase to decrease frequency deviations. As such, together with the fact that they do not decrease deviations sufficiently, these parameters are unfit as a frequency control method.

VI. DISCUSSION

A. Analysis of FFR with Battery Support

For the model used in Section III, the FFR control method using batteries is applied to regulate the system and provide better frequency stability. For this system, FFR serves as a complement to FCR-D and provides a faster activation for major frequency deviations compared to FCR-D [7]. The fast activation of batteries makes them a more suitable source of FFR than control methods that use governor control to increase production in power plants such as hydropower. The attempted, alternative control method analysed in Section V proved unsuccessful in providing a fast enough increase in generated power to handle disturbances in the low inertia case. However, the battery control method does come with a few limitations.

In Fig. 8, it can be seen that the battery support provides sufficient power to minimise maximum frequency deviations for various disturbances and inertia cases. Nonetheless, if the disturbance is small enough and the inertia sufficient, overestimation of the required power output of the battery during the disturbance is possible. By injecting too much power into the system, it is possible to overshoot the frequency levels past the nominal value. Additionally, by overusing the capabilities of the battery support, the battery will expire faster and have increased running costs. Hence, an additional

control algorithm for the battery power output could prove to be essential for prolonging the lifespan of the system and lowering its costs.

B. Battery Storage and Cost Analysis

There are several ways to implement battery units used for frequency support with FCR and FFR with batteries. In [21], it is said that the most suitable solution for battery support is Lithium-ion batteries given their high efficiency. The batteries can either be newly produced specifically for the storage system or repurposed from electric vehicles, which is mentioned in [22]. Furthermore, if the capacity of the batteries is big enough, they can provide both FFR and be used as a BESS to store and supply power on a greater scale for a longer duration of time. This helps to further prevent blackouts in the electricity grid.

The study in [23] provides a 2021 yearly report on the cost of different energy technologies, such as large-scale energy storage systems, in the US market. These storage systems consist of Lithium-ion batteries and flow batteries with an approximate life cycle of 20 years. For a system with a storage capacity of 100 MW, the storage duration will last for an hour when activated and will consume approximately 31500 MWh of energy annually. For a battery with a capacity of 100 MW, the yearly costs would approximately be 600 to 1000 SEK for every kW each year or 1600 to 2800 SEK for every MWh each year.

The battery used in this context of work has a size of 430 MW. For one instance of activation, during a frequency deviation, the battery will consume 5.12 GWs which corresponds to 1.42 MWh. This should give a cost of 9800 to 17100 SEK for the activation with the assumption that the cost is increased proportionally with the size of the battery. This cost might seem a bit extensive for the battery implementation. However, the source data is modelled for a battery that is used for an hour daily, while the battery used for frequency support might not be activated at all on some days. Hence, if the battery size is not proportional to its running cost and if it is not activated for an hour each day, these costs might be lower. Nevertheless, larger batteries will come with an increased cost for the maintenance of the grid, which needs to be considered when planning its operation.

C. Alternate Designs to Reduce Frequency Deviations

In [9], an alternative control method to FFR is discussed using power converters, coupled with a supplemental energy storage device, to simulate a Virtual Synchronous Generator (VSG) which provides synthetic inertia. The VSG control adopts a control algorithm to simulate the usage of a physical synchronous generator. Apart from providing synthetic inertia, the control method also allows switching between high and low inertia levels. This is useful since the frequency stability only relies on the inertia levels during IFR and a few moments afterwards, while lower levels of inertia provide faster stabilisation [9]. Comparing VSG to the FFR controller, the key differences are that FFR along with FCR-D provides a

fast activation in case of disturbance while VSG improves the settling time for the SSFD.

In [10] Variable Speed Hydropower (VSHP) is mentioned as an alternative to regular hydropower generation. VSHP utilises power converters connected to the rotating turbines that are coupled with water reservoirs to control the water flow in addition to the turbine governor found in regular hydropower. In case of a disturbance, the power converters can quickly adjust the water flow to change the rotational speed of the turbines, which regulates the power to and from the electricity grid. This method does not provide any inertia, but it does provide synthetic inertia from the power converter and it also operates faster compared to using governor control. Benefits of using VSHP include lowered frequency deviations, reduced power oscillations, and increased Critical Fault Clearing Time (CFCT). CFCT is the maximum duration for which the disturbance can occur without the system losing stability. In [10], it is further mentioned that VSHP already exists in Norway's transmission system, which could make it easier to adapt for other Nordic TSOs.

In [24], a control method applied to HVDC-based systems is implemented to provide FFR to a low inertia system. By changing the active power output almost instantaneously, HVDC lines are argued to better enhance frequency stability than control methods where the injection of power is gradual. The method estimates the size of the disturbance and the HVDC response is adapted accordingly to secure an acceptable frequency. Using HVDC as a source of frequency support may have the additional benefit of reduced costs for the Nordic TSOs, as presented in [25].

D. Future Research

The linearised system model can be further improved to accurately reflect the Nordic power system by adding RES and other energy sources. Furthermore, a battery control method including the possibility to regulate the power output depending on the size of the power disturbance can be applied for a more efficient system.

VII. CONCLUSION

The reduced amount of system inertia following the increased integration of power electronics-based RES is seen to impact the frequency stability of a power system. By analysing the frequency response of a linearised power system model after a disturbance, it is concluded that reduced inertia increases RoCoF and maximum IFD. This could make a power system more vulnerable to disturbances, as high values of RoCoF and maximum IFD may lead to system separation and power outages. The amount of inertia is not the only factor affecting the frequency response, as parameters related to the implementation of FCR in the system model are also seen to affect RoCoF, maximum IFD, and SSFD. For the investigated low inertia scenario, the implemented FCR-D proved not to be sufficiently fast to ensure the stability of frequency after a disturbance. To support the frequency in case of a disturbance, supplementary battery control was included in the system model. After its implementation, the frequency was

successfully contained within the allowed interval. However, the battery control method could be developed to adjust its power output based on the size of the disturbance.

ACKNOWLEDGEMENT

The authors would like to thank Danilo Obradović for consistently supporting the project with his expertise, guidance, and participation in all scheduled project meetings throughout the project.

REFERENCES

- [1] Nordic Energy Research. (2018) The nordics: Nordic electricity generation and trade, 2017. [Online]. Available: <https://www.nordicenergy.org/figure/nordic-electricity-generation-and-trade-2017/>
- [2] Nordic-TSOs. (2016, Aug.) Challenges and opportunities for the nordic power system. [Online]. Available: <https://www.fingrid.fi/globalassets/dokumentit/fi/yhtio/ki-toiminta/report-challenges-and-opportunities-for-the-nordic-power-system.pdf>
- [3] E. Ørum *et al.* (2015) Future system inertia. [Online]. Available: https://eepublicdownloads.entsoe.eu/clean-documents/Publications/SOC/Nordic/Nordic_report_Future_System_Inertia.pdf
- [4] M. Ghandari, "Stability of power systems: An introduction," KTH Royal Institute of Technology, Stockholm, Sweden, 2018.
- [5] Svenska kraftnät. (2015) Natutvecklingsplan 2016–2025. [Online]. Available: <https://www.svk.se/siteassets/om-oss/rapporter/2016/natutvecklingsplan-2016-2025.pdf>
- [6] L. Wingren and J. Johnsson, "Battery energy storage systems as an alternative to gas turbines for the fast active disturbance reserve," Ma. thesis, Lund University, Lund, Sweden, 2018.
- [7] ENTSO-E. (2019, Dec.) Fast frequency reserve – solution to the nordic inertia challenge. [Online]. Available: https://www.statnett.no/globalassets/for-aktorer-i-kraftsystemet/utvikling-av-kraftsystemet/nordisk-frekvensstabilitet/ffr-stakeholder-report_13122019.pdf
- [8] E. Bergvall and A. Bonetti, "Frequency Stability in Future Low Inertia Power Systems With Battery Support," Bsc. thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2021.
- [9] L. Huanjing, Z. Li, Z. Tong, and Z. Liang, "Frequency regulation strategy for dfig combining over-speed control and adaptive virtual inertia," in *2019 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*. Singapore: IEEE, 2019, pp. 1663–1666.
- [10] T. I. Reigstad and K. Uhlen, "Variable speed hydropower for provision of fast frequency reserves in the nordic grid," *IEEE Transactions on Power Systems*, vol. 36, no. 6, pp. 5476–5485, 2021.
- [11] Fingrid. (2022, Apr.) Nordic power system and interconnections with other systems. [Online]. Available: <https://www.fingrid.fi/en/grid/power-transmission/nordic-power-system-and-interconnections-with-other-systems/>
- [12] ENTSO-E. (2022, Apr.) Entso-e member companies. [Online]. Available: <https://www.entsoe.eu/about/inside-entsoe/members/>
- [13] ENTSO-E. (2021, Feb.) Operational limits and conditions for mutual frequency support over hvdc. [Online]. Available: https://eepublicdownloads.entsoe.eu/clean-documents/SOC%20documents/Operational_Limits_and_Conditions_for_Mutual_Frequency_Support_over_HVDC_Report.pdf
- [14] ENTSO-E. (2021, Mar.) Technical Requirements for Frequency Containment Reserve Provision in the Nordic Synchronous Area. [Online]. Available: <https://www.statnett.no/globalassets/for-aktorer-i-kraftsystemet/utvikling-av-kraftsystemet/nordisk-frekvensstabilitet/fcp-pilot-2021/vedlegg-2---draft-2021---technical-requirements-for-fcr-in-the-nordic-synchronous-area---pilot.pdf>
- [15] Nordenergi. (2021, May) Study on opportunities and barriers to electrification in the Nordic region. [Online]. Available: https://www.danskenergi.dk/sites/danskenergi.dk/files/media/dokumenter/2021-05/Study-on-opportunities-and-barriers-to-electrification-in-the-Nordic-region_2.pdf
- [16] ENTSO-E. (2021, Jan.) Technical requirements for fast frequency reserve provision in the nordic synchronous area - external document. [Online]. Available: <https://www.statnett.no/globalassets/for-aktorer-i-kraftsystemet/marked/reservemarkeder/ffr/technical-requirements-for-ffr-v1.1.pdf>
- [17] S. Ming, X. Gangwen, C. Lei, L. Yuming, L. Xiaoju, and M. Yong, "Study on the necessity and role of inertia in power system frequency response," in *2020 IEEE 4th Conference on Energy Internet and Energy System Integration (EI2)*. New Jersey: IEEE, 2020, pp. 155–159.
- [18] D. Zografos, "Power system inertia estimation and frequency response assessment," PhD dissertation, KTH Royal Institute of Technology, Stockholm, Sweden, 2019.
- [19] T. Fei, M. Yunfei, J. Hongjie, W. Jianzhong, Z. Pingliang, and S. Goran. (2016) Challenges on primary frequency control and potential solution from evs in the future gb electricity system. [Online]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S0306261916307280>
- [20] 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)*. New Jersey: IEEE, 2016, pp. 1–6.
- [21] T. Mäkinen, A. Leinonen, and M. Ovaskainen, "Modelling and benefits of combined operation of hydropower unit and battery energy storage system on grid primary frequency control," in *2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I CPS Europe)*. New Jersey: IEEE, 2020, pp. 1–6.
- [22] IEEE, "IEEE guide for the characterization and evaluation of lithium-based batteries in stationary applications," *IEEE Std 1679.1-2017*, pp. 1–47, 2018.
- [23] Lazard. (2021, Oct.) Lazard's leveled cost of storage analysis—version 7.0. [Online]. Available: <https://www.lazard.com/media/451882/lazards-levelized-cost-of-storage-version-70-vf.pdf>
- [24] J. Stojković, A. Lekić, and P. Stefanov, "Adaptive control of hvdc links for frequency stability enhancement in low-inertia systems," *Energies*, 2020.
- [25] A. Tosatto, M. Djokas, T. Weckesser, S. Chatzivasileiadis, and R. Eriksson, "Sharing reserves through HVDC: Potential cost savings in the nordic countries," *IET Generation, Transmission Distribution*, 2020.

Small Signal Stability of Power Systems With Increased Converter-Based Power Production

Gustav Kjellson and Josefin Gustafsson

Abstract—The aim of this project is to analyze how increasing the share of converter-based power production in a power system affects the small signal stability. The project also aims to stabilize systems using Power System Stabilizers (PSSs), and examine the effect of two different types of power electronic converters: grid following (GFL) and grid forming (GFM). To do this, a short-circuit fault is simulated in a two-area four-machine power system using DigSILENT PowerFactory. The project examines how replacing one of the synchronous machines with a converter-based power source affects the system's stability. Modal analysis is used to assess the small signal stability, as well as to tune the PSS. In the project PSSs are successfully used to stabilize systems, both with and without converter-based power production. The study also finds that implementing power production using GFL worsens the small signal stability while GFM improves it.

Sammanfattning—Syftet med det här projektet är att analysera hur den ökande andelen konverter baserad elproduktion påverkar *small signal stability* i ett elkraftsystem. Projektets syfte är också att stabilisera system med *Power System Stabilizers* (PSSs), och undersöka påverkan av två typer av effektelektronik: *grid following converter* (GFL) och *grid forming converter* (GFM). Det görs genom att simulera ett kortslutningsfel i ett elkraftsystem med DigSILENT PowerFactory. I projektet undersöks hur systemets stabilitet påverkas av att ersätta en synkrogenerator med effektelektronik-kopplad elproduktion. Egenvärdesanalys används för att undersöka *small signal stability* och implementera PSS. I projektet lyckas stabiliseringen med PSS för både systemet med och utan effektelektronik-kopplade energikällor. Studien visar också att genom implementeringen av energikällor med GFL försämras stabiliteten, medan GFM förbättrar den.

Index Terms—Small signal stability, Power system stabilizer, Renewable energy, Grid following converters, Grid forming converters.

Supervisors: Angel Clark and Merhdad Ghandhari

TRITA number: TRITA-EECS-EX-2022:141

I. INTRODUCTION

The severe risks of climate change for both society and ecosystems are described in the IPCC report [1]. The report explains the importance of staying under the 1.5 °C global warming target, which will require greatly reducing greenhouse gas emissions. The actions taken to tackle climate change in the near future will be crucial, determining the severity of the consequences affecting humans and ecosystems. In [2] it is presented that 73.2 % of the world's greenhouse gas emissions in 2020 came from the energy sector, which includes electricity, heat and transportation. To reduce these emissions, solutions within several different areas are needed. One such solution is increasing the penetration of renewable

energy sources in power systems. According to [3] the investment costs for implementation of wind and photovoltaic (PV) solar are decreasing, enabling the expansion of these power sources. However, there are challenges associated with achieving resilient electrical power systems containing a large share of these renewable energy sources.

Wind and PV solar are converter-based renewable energy sources (CBRES). In [4] it is described that wind and PV solar differs from hydro, nuclear and fossil fueled power generation in two important ways. Firstly, the power generation is variable since it depends on weather conditions. Secondly, they are connected to the grid with power electronic converters, while conventional energy sources are connected to the grid with synchronous generators (SGs). As explained in [4], SGs have an electromagnetic coupling to the grid which means the rotational energy in their turbines provides the system with inertia. This characteristic of synchronous generators is heavily used in the control of electrical power systems. Therefore, when converter-based power sources replaces synchronous generation it will change the behavior and control method of electrical power systems.

Achieving stable power systems resilient to disturbances, and with high levels of CBRES, is already a problem around the world today. An example is the all-island Irish transmission system, which according to [5] has a set limit of 65% instantaneous converter-based power production. If this limit is exceeded it can lead to stability issues. The instantaneous power production from their wind farms have passed this percentage, and today this limit restricts their utilization of wind energy. As presented in [4], instantaneous power production can be much higher than the annual average. For smaller AC power systems with high levels of CBRES this sort of problem of momentarily exceeding these limits is already seen today. With the noticeable increase of wind and PV solar, finding feasible solutions to these challenges will only become more important. To do this, an understanding of how the implementation of CBRES affects power systems is needed.

The aim of this project is to examine the effects of implementing CBRES into an electrical power system, considering both the effects on the system and on the control options. This is done by comparing how the same two-area power system, with and without penetration of wind power, responds to a short-circuit fault and how it can be stabilized. The rotor angle stability is examined and improved. This is done with power system stabilizers (PSSs), which as described in [6] are control components added to the synchronous generators. In addition, two different methods of converter control are compared. First a grid-following converter (GFL), and then a grid-forming

converter (GFM), are used to connect the wind farm to the grid. The main difference between them is described in [7]: a GFM can provide a frequency and voltage reference to the system which the GFL does not. This project focuses on the effects of replacing synchronous generators with converter-based power sources. Thus, the aspect of variability in power generation caused by the wind farm is not considered.

Previous work, such as [8], has shown that implementing high shares of CBRES using GFL can lead to a worsened stability in a power system. In [8] it is also presented that GFM is a possible solution for overcoming these issues. This project contributes to a better understanding of the impact on stability of GFL and GFM, as well as on whether PSSs can be used to stabilize power systems with CBRES.

Section II includes definitions, descriptions of control methods and the theory behind calculation methods used in the project. Section III presents the system model, simulations and calculation methods. Section IV presents the project's results and a short analysis of them. Section V includes a deeper analysis and comparison of the results, alternative methods and future research. The projects results are summarized in the Section VI.

II. THEORY

A. Rotor angle stability

Rotor angle stability is described in [6] as the ability for synchronous machines in the same system to remain in synchronization with each other. When a disturbance affects a power system this can lead to loss of synchronization between the synchronous generators in the grid. This means they will start rotating with different generator speeds, and therefore the angle between them will change with time. Severe disturbances leading to a large deviation in the rotor angle will cause the synchronous generators to lose synchronization, which can lead to disconnection of singular machines or areas in the system. A common problem is inter-area oscillations where generators in separate areas of a system oscillate against each other.

There are two types of rotor angle stability: transient and small signal stability. The difference between these two types of stability is described in [6]. Transient stability is the system's ability to stabilize after large disturbances. Small signal stability is the system's ability to stay synchronized after small disturbances, which for example can be minor short-circuit faults, variations in power production or consumption. Small signal stability control can be applied when the disturbances occurring are small enough that a linear approximation of the system's equations will give a reasonable analysis of the system. If this is not the case, non-linear transient stability methods need to be used.

B. Modal Analysis

One method for analyzing small signal stability is modal analysis. Since the goal of the project is to analyze small signal stability, it can be assumed that studying a linear approximation of the system is a valid approach. Modal analysis looks at properties of the linearized system's eigenvalues. As described

in [6], the dynamics of a multi machine power system can be described by the following system of equations.

$$\dot{x} = f(x, y) \quad (1)$$

$$0 = g(x, y) \quad (2)$$

Equation 1 is a set of nonlinear differential equations describing generator dynamics and Equation 2 is a set of algebraic equations based on Kirchhoff's current law at each terminal. The variable x is a set of the system's state variables, such as generator speeds, and y is the set of algebraic variables, such as terminal voltages. The system can be linearized around a stable operating point (x_0, y_0) in order to then calculate and analyze the system state matrix's eigenvalues. The linearized system is given by the following equations

$$\Delta \dot{x} = f_x \Delta x + f_y \Delta y \quad (3)$$

$$0 = g_x \Delta x + g_y \Delta y \quad (4)$$

where f_x , g_x , f_y , g_y are the jacobian matrices of $f(x, y)$, $g(x, y)$ with respect to x or y , evaluated at the stable operating point (x_0, y_0) . Equations 3 and 4 can be manipulated to get the following equation for the overall system state matrix A .

$$\Delta \dot{x} = (f_x - f_y(g_y)^{-1}g_x)\Delta x = A\Delta x \quad (5)$$

The linearized system's eigenvalues can then be calculated by finding all λ that satisfy

$$\det(A - \lambda I) = 0 \quad (6)$$

where \det refers to the determinant and I is the identity matrix. The eigenvalues of the linearized system's state matrix are referred to as modes of the system. The modes are given by the following equation.

$$\lambda = \sigma \pm j\omega \quad (7)$$

They will have a real part σ and an imaginary part ω . For the linearized system to be small signal stable all modes must have a negative real part. A mode with a nonzero complex part will also need a positive damping ratio. The complex part of a mode contributes to oscillations but a positive damping ratio will lead to stable oscillations with a decreasing amplitude. The modes damping ratio ζ can be calculated with the following equation.

$$\zeta = \frac{-\sigma}{\sqrt{\sigma^2 + \omega^2}} \quad (8)$$

When assessing small signal stability, using modal analysis, it is also of interest to look at which state variables contribute the most to the placement of a certain mode. This is done by calculating the participation factor p_{ki} which shows the relative participation of the state variable k in the placement of the i -th mode λ_i . The participation factor is calculated using the equation

$$p_{ki} = \frac{\partial \lambda_i}{\partial a_{kk}} \quad (9)$$

where a_{kk} is the element in the k -th row and k -th column of the system state matrix A . Finding the state variables with the highest participation factor helps to determine what part of the system affects the mode the most.

C. Synchronous generators

A synchronous generator converts mechanical power from a turbine to electrical power. The following equation, shown in [6], describes the three phase active power production.

$$P_e = 3 \frac{EU}{X_s} \sin \delta \quad (10)$$

where E is the generators internal phase voltage, U is the grid phase voltage, δ is the phase difference between E and U , and X_s is the generators internal reactance. Studying the rotation of synchronous machines when affected by a disturbance is an important aspect of understanding the electromechanical dynamics of a power system. A change in P_e affects the rotation of the synchronous generator as shown in the following equation, known as the swing equation.

$$\dot{\omega} = \frac{1}{M}(P_m - P_e) \quad (11)$$

ω is the deviation between the rotor speed and the system's synchronous speed in per unit (p.u). P_m and P_e are the mechanical and electrical active power in p.u. The constant M is given by

$$M = \frac{2HS_{ng}}{\omega_s S_{base}} \quad (12)$$

where H is the inertia constant measured in seconds. S_{ng} is the generator's rated three-phase complex power, S_{base} is an arbitrary three-phase base power used to convert P_m and P_e to per unit and ω_s is the electrical synchronous speed. As all variables in M except for H are constant the change in M will only depend on the inertia constant. Thus, the swing equation describes the dynamics of a synchronous generator depending on the machines inertia, deviations in mechanical and electrical power and the rate of change of the rotor speed. When a disturbance causes a deviation between the mechanical and electrical power a high inertia will reduce the rotor speed's acceleration and therefore contribute to rotor angle stability.

D. Excitation systems

An exciter supplies the current to the rotor which determines a synchronous generator's terminal voltage. Excitation systems therefore include an automatic voltage regulator (AVR) that controls the generator's output voltage by changing this current. The AVR used in this project is the ESAC4A model, a simplified block diagram of the transfer function is shown in Fig. 1. The voltage measurement is shown by u , u_{ref} is the reference voltage and u_{pss} is a supplementary voltage signal explained below.

While the AVR improves power system stability, [6] describes that an excitation system with a high gain can cause poorly damped electromechanical oscillations in a multi-machine system. This can lead to rotor angle instability if the

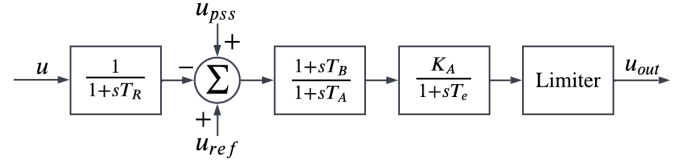


Fig. 1. Block diagram of the ESAC4A exciter model.

oscillations are not properly damped. To negate this an extra form of control called a power system stabilizer (PSS) can be utilized. The PSS improves the damping ratio by providing a supplementary voltage signal to the AVR to compensate for the high gain. In this project the PSS-STAB1 model is used, the block diagram of the transfer function is shown in Fig. 2. It has the generator speed ω as the input signal and the supplementary voltage signal u_{pss} as the output. The first block is a high pass filter with a static gain, the purpose of this filter is to tune out steady state changes in the generator speed. The second and third block are two lead lag blocks that shift the phase in order to achieve a positive contribution to the damping ratio. The last block is a limiter that sets a maximum allowed value u_{lim} in p.u for the output signal. In this project only the gain value K is tuned when implementing PSSs. The other constants are left at the default values which are presented in Table I.

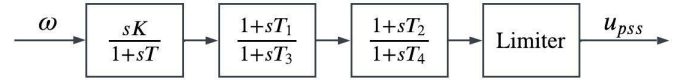


Fig. 2. Block diagram of the PSS-STAB1 model.

TABLE I
DEFAULT CONSTANTS FOR THE PSS-STAB1 MODEL

Constant	Value
T	10 [s]
T_1	0.5 [s]
T_2	0.5 [s]
T_3	0.05 [s]
T_4	0.05 [s]
u_{lim}	0.03 [p.u]

E. Converter control

Energy sources such as wind and solar PV are connected to the grid by power electronic converters. As mentioned in the introduction there are different types of converters, they are divided into two main categories: grid following converters (GFL) and grid forming converters (GFM). According to [7] the difference between a GFL and GFM can be explained by that the GFL acts as a current source and the GFM as a voltage source. This means the GFL controls the converter current while the GFM controls the converter voltage. This results in different responses to disturbances. In Fig. 3 the GFL's and GFM's response to a change in grid voltage show their different behavior. The GFL preserves the current's phase and

magnitude, leading to an alteration of the converter voltage. For the GFM the internal voltage is not affected by the disturbance, instead the current is adjusted to match the grid conditions. According to [9] this makes the GFM more similar to a synchronous generator as it provides a steady voltage and frequency to the grid. The GFM's hardware limits the magnitude of the converter current that the GFM can supply. In order to supply the current for maintaining grid voltage there also needs to be a sufficient power reserve.

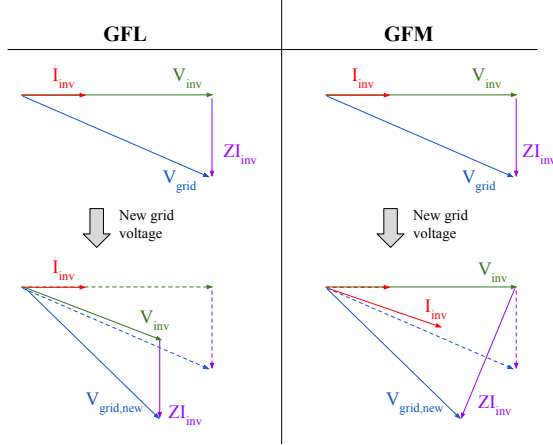


Fig. 3. Phasor diagram showing the difference between how a GFL and a GFM responds to a grid voltage deviation.

Another difference described in [7] is that the GFL's current control follows the grid conditions while the GFM sets its own reference voltage. This means the GFL needs to measure the angle of the grid voltage in order to calculate a new set point for the current control. This is done using a control system called a phase-locked loop (PLL) which has been shown to be unstable in weak grid scenarios. Thereby the GFL control needs a stronger system to work sufficiently, as described in [9]. On the other hand, the GFM provides a reference to the grid and does not depend on measuring the grid voltage. This characteristic makes it similar to a SG and therefore the GFM can function in a weak grid.

There are different types of GFM, as described in [10]. The type used in this project is the virtual synchronous machine (VSM). The main concept of a VSM is that it emulates the features of a synchronous machine, providing the system with virtual inertia as explained in [10] and [11]. This can be done in different ways, for example by using the swing equation. Fig. 4 shows a simplified block diagram for the VSM used in the project.

The virtual rotor's inertia is represented by the mechanical time constant $T_a = 2H$. D_p is the damping coefficient, p_{set} is the active power set point and p_{mea} is the measured active power. ω_{VSM} is the VSM's virtual rotor speed, while ω_{set} represents the speed set point. The integration of ω_{VSM} gives the voltage angle θ . $\omega_{set,N}$ converts the angle to per unit. The VSM is only one part of the power converter control. The voltage amplitude, for example, is controlled in a separate control structure within the converter.

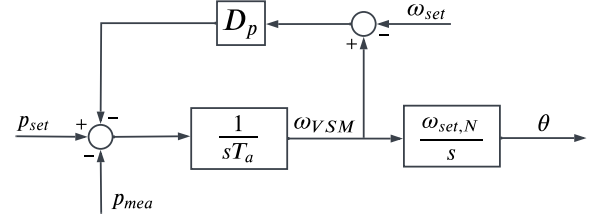


Fig. 4. Simplified block diagram of the VSM model.

III. METHODOLOGY

A. Power System Model

The system models and simulations are done in DigSILENT PowerFactory 2022, which is a power system analysis software. In addition MATLAB is used for graphical plots and calculations. The models are based on a two-area four-machine base model and three different versions of the system model are analyzed. All systems have a frequency of 50 Hz.

The first version, the SG system, models a system with only conventional energy sources connected to the grid by synchronous generators (SGs). The system uses a sixth-order synchronous generator model. Fig. 5 shows the system model. Each of the two areas includes one general load and two power sources. The SGs at terminals one to four are from now on referred to as G1 to G4. The two areas, left and right side, are connected to each other at terminal eight. The generators are connected to the grid with three-phase transformers. At terminal seven and nine there are general loads and also shunt filters with the rated reactive power of 200 MVAR at terminal seven and 350 MVAR at terminal nine. In Table II the set point for the active and reactive power from each generator and to each load at steady state is presented. All the generators have a rated complex power of 900 MVA.

In the second and third version of the system model the SG at terminal two is replaced by a wind farm. For the second version, the GFL system, the wind farm is connected to the grid with a GFL and for the third version, the GFM system, it is connected with a GFM instead. Fig. 6 shows the model of the wind farm connected at terminal two. The wind farm contains 250 parallel units of wind turbines, each connected to a three phase transformer. In the model the wind farm's power production is constant and equal to the production of the replaced SG. Therefore the system models do not take the variability of wind power production into consideration but only show the difference between converter-based power production and synchronous generators.

An RMS simulation of a solid short-circuit fault is applied at terminal eight after 2.0 seconds and cleared after 2.1 seconds in all systems. The same disturbance is used for all systems in order to compare their response.

B. Implementing and Tuning the PSS

To stabilize the systems, PSS are implemented in the SG and the GFL systems. They are not used in the GFM system. The PSS are implemented and tuned based on modal analysis.

TABLE II
ACTIVE AND REACTIVE POWER AT STEADY STATE

Generator/Load	P [MW]	Q [MVAR]
Generator 1	700	185
Generator 2	700	235
Generator 3	719	176
Generator 4	700	202
Load 7	967	100
Load 9	1767	100

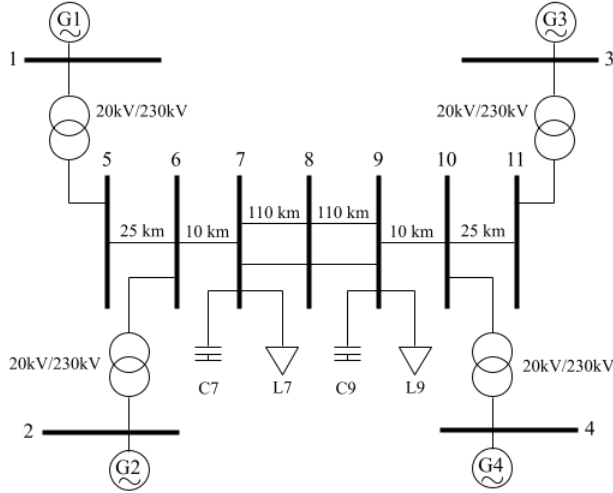


Fig. 5. Two-area four-machine base model of a system with only SGs.

In the SG system there are four generators where a PSS could potentially be implemented, and in the GFL system there are three. In order to identify the optimal placement of the PSS the participation factors for the mode with the minimum damping ratio must be calculated. The power system's modes, their damping ratios and the corresponding participation factors are calculated in PowerFactory. The PSS should then be implemented at the generator whose state variables have the highest relative participation for the least damped mode. This is done in order to improve the system stability as much as possible. If it is considered that more than one PSS is necessary the same procedure can be repeated after tuning the first one.

For the PSS to improve the system stability as much as possible the constants need to be tuned. This project only tunes the static gain while the remaining constants are set to the model's default values, which are shown in Table I. In order to tune the gain, the minimum damping ratio is calculated and compared for different gain values. By calculating the damping ratio for a span of different gain values the optimal gain can be found with sufficient accuracy for the purpose of the project.

In order to compare the stability of the different systems the settling time of the rotor angles after the short-circuit fault is calculated. The settling time is defined as the time until all rotor angles lie within 0.5 degrees of their final steady state value. This allows for fair comparisons between the SG, GFL and GFM systems.

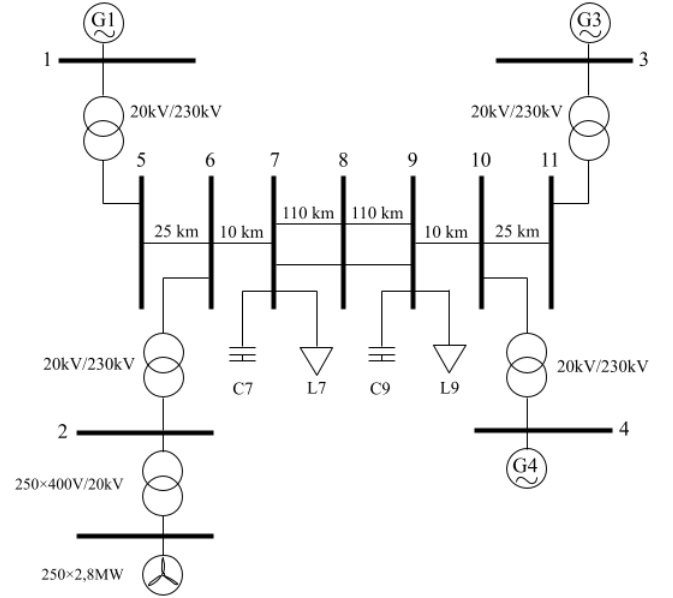


Fig. 6. Two-area four-machine base model with a wind farm replacing the SG at terminal 2.

IV. RESULTS

A. SG system without PSS

First the response of the SG system, shown in Fig. 5, without PSSs is analyzed. Fig. 7 shows the rotor angle deviation from the steady state value for all four generators. G1 is the reference machine which means the plots show the change in rotor angle compared to the angle of G1. Fig. 8 shows the terminal voltages for all four generators during the fault simulation. Fig. 7 shows that the system is small signal unstable as the rotor angles oscillate with a growing amplitude. There are also visible inter-area oscillations with G3 and G4 oscillating against G1 and G2. The instability of the system can also be seen by looking at the properties of the least damped mode, which is shown in Table III where f_D is the damped frequency and ζ_{min} is the mode's damping ratio.

TABLE III
THE LEAST DAMPED MODE IN THE SG SYSTEM WITH NO PSS

Mode [s^{-1}]	f_D [Hz]	ζ_{min} [%]
$0.14 \pm j3.37$	0.54	-4.02

When the short-circuit fault occurs at 2.0 seconds there is a sharp voltage drop at all terminals visible in Fig. 8. This results in a sudden loss of active power production according to Equation 10. Assuming the supplied mechanical power is constant during the disturbance this will lead to an acceleration of the generator speeds according to Equation 11. The acceleration is proportional to the loss of active power at each generator terminal and so the generator speeds will no longer be equal to each other. This leads to deviations in the relative rotor angle. As the fault is cleared the terminal voltages are restored but the high excitation gain causes voltage oscillations with increasing amplitude due to the negative damping ratio.

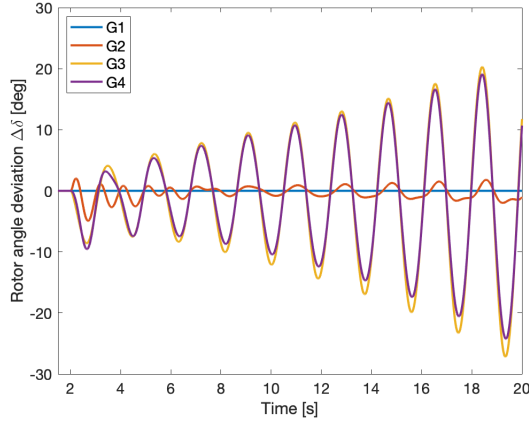


Fig. 7. Rotor angles of G1-G4 from the fault simulation for the SG system.

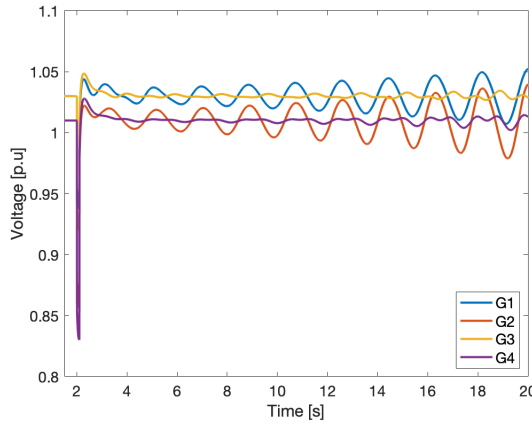


Fig. 8. Voltages at terminal 1-4 from the fault simulation for the SG system.

This leads to oscillations in the active power and generator speeds which in turn leads to oscillations in the rotor angles as seen in Fig. 7.

B. SG system with PSS

In order to stabilize the system PSSs are implemented. The mode in Table III, with the minimum damping ratio in the SG system, has high participation factors for G3's speed and angle. Therefore the first PSS is implemented at G3. Fig. 9a shows the minimum damping ratio in the system with different gain values for the PSS at G3 (K_{G3}). The optimal K_{G3} was found to be 214. The properties of the least damped mode in the system with one PSS implemented are shown in Table IV.

TABLE IV
THE LEAST DAMPED MODE IN THE SG SYSTEM WITH ONE PSS

Gain	Mode [s^{-1}]	f_D [Hz]	ζ_{min} [%]
$K_{G3} = 214$	$-0.38 \pm j6.66$	1.06	5.64

The system is stable with just one PSS implemented, but to improve the settling time a second PSS is implemented. With K_{G3} set at the optimal value the limiting mode from Table IV has high participation factors for G2's speed and angle. Therefore, the next PSS is implemented at G2. The

blue curve in Fig. 9b shows the minimum damping ratio in the system with different gain values for the PSS at G2 (K_{G2}) while K_{G3} is set to 214. Fig. 9b shows that implementing a PSS at G2 greatly enhances system stability as the minimum damping ratio improves drastically. The optimal gain for G2 is found to be roughly 125.

A gain of 214 at G3 is very high and unrealistic. A high gain can also lead to unintentionally pushing other modes to instability. Therefore K_{G3} is lowered to the same value as K_{G2} . This will worsen the minimum damping ratio but Fig. 10 shows that the system response is still sufficient.

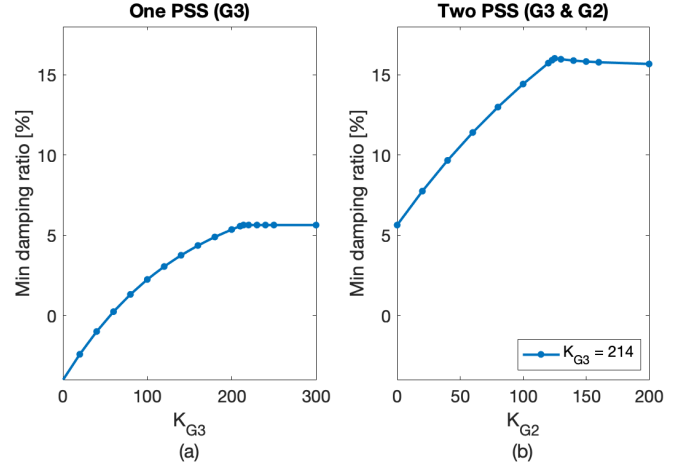


Fig. 9. a) The linearized system's minimum damping ratio for different values of K_{G3} . b) The linearized system's minimum damping ratio for different values of K_{G2} with a PSS at G3.

Fig. 10 shows the rotor angles of all four generators with two PSS implemented. Note that Fig. 10 shows the response with the lower gain value of 125 at G3. The system is now stable, as all four generators return to the steady state value within the simulation time. The settling time is calculated for both the optimal gain and lower gain alternative, which is shown in Table V along with the minimum damping ratio. The settling time is actually lower for the system with lower gain at G3, despite having a worse minimum damping ratio, although this is largely due to the chosen margin for the settling time (± 0.5 degrees). The lower gain alternative has small oscillations for slightly longer than the optimal gain alternative but they are so small they do not affect the settling time.

TABLE V
STABILIZED SG SYSTEM COMPARISON

Gain	T_s [s]	Mode [s^{-1}]	f_D [Hz]	ζ_{min} [%]
$K_{G3} = 214$ $K_{G2} = 125$	9.92	$-0.90 \pm j5.57$	0.89	16.02
$K_{G3} = 125$ $K_{G2} = 125$	9.56	$-0.33 \pm j2.82$	0.43	11.87

C. GFL system without PSS

In the GFL system the synchronous generator at terminal two is replaced by a wind farm as shown in Fig. 6. The wind

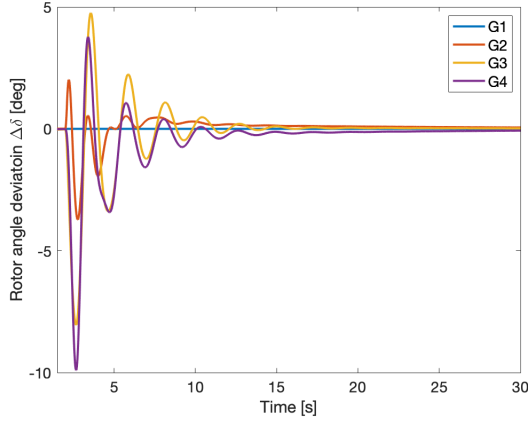


Fig. 10. Rotor angles of G1-G4 from the fault simulation for the stabilized SG system with $K_{G3} = K_{G2} = 125$.

farm is connected to the grid with a grid following converter. The system is simulated with the same short-circuit fault as the SG system. Fig. 11 shows the rotor angles of G1, G3 and G4 and Fig. 12 shows the terminal voltages at terminals one to four. Note that since G2 has been replaced by the converter connected wind farm there is no rotor angle to plot for that terminal.

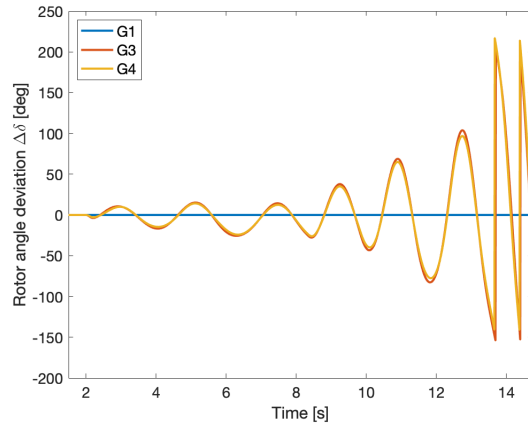


Fig. 11. Rotor angles of G1, G3 and G4 from the fault simulation for the GFL system.

The system response is unstable and the rotor angle oscillations grow faster than in the unstable SG system. The deviation from the steady state values are larger for both the rotor angles and the terminal voltages, when compared to the unstable SG system. At roughly 13 seconds the two areas separate and lose synchronism as G1 starts to accelerate rapidly. The worsened stability of the system is also shown by the least damped mode having a worse damping ratio than in the SG system. The properties of the least damped mode are shown in Table VI.

TABLE VI
THE LEAST DAMPED MODE IN THE GFL SYSTEM WITH NO PSS

Mode [s^{-1}]	f_D [Hz]	ζ_{min} [%]
$0.23 \pm j2.96$	0.47	-7.62

Looking at Fig. 12 it is clear that the voltage oscillations are

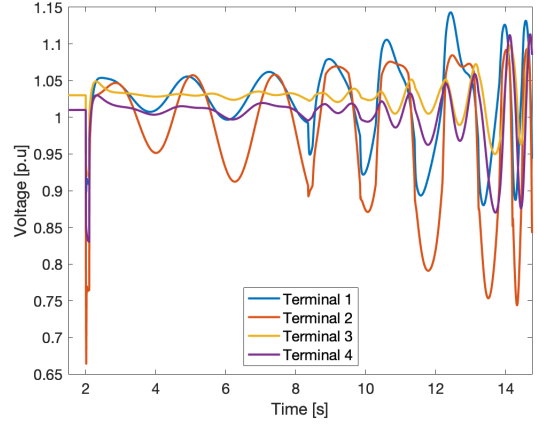


Fig. 12. Voltages at terminal 1-4 from the fault simulation for the GFL system.

largest at terminal two where the wind farm is located. This makes sense as the grid following converter has no voltage control. Just as in the SG system the voltage oscillations cause active power oscillations leading to changes in the generator speed according to Equation 11. The system is affected by the loss of inertia in the left side of the system as Equation 11 also shows that a decrease in inertia will lead to a greater generator acceleration for a given loss of electric power. This means that G1's speed is more sensitive to active power oscillations since it is in the left system area with the wind farm.

D. GFL system with PSS

Following the same procedure as in the SG system, PSSs are implemented to stabilize the system. The mode with the minimum damping ratio in the system has high participation factors for G3's speed and angle. Therefore the first PSS is implemented at G3 with the gain K_{G3} . Increasing the gain to high values does not result in a stable system as one PSS is not enough to stabilize the GFL system, which it was for the SG system. With one PSS installed at G3 the mode with the minimum damping ratio has large participation factors for G4's speed and angle. However, implementing a second PSS at G4 does not greatly enhance system stability, while implementing one at G1 does. Since it is mainly inter-area oscillations that need to be damped one PSS is implemented in each area of the system. Due to the optimal gain for the PSS at G3 being unreasonably high, different gain combinations for both G1 and G3 were tested, see Table VII.

TABLE VII
MINIMUM DAMPING RATIO [%] FOR COMBINATIONS OF K_{G1} AND K_{G3}

$K_{G1} \rightarrow$ $K_{G3} \downarrow$	0	50	100	150	200
50	-5.35	-2.20	2.48	4.06	5.07
100	-4.11	2.97	6.55	8.59	9.89
150	-3.49	5.20	9.52	11.98	13.54
200	-3.23	6.80	11.76	14.10	14.02

The optimal combination is K_{G1} at 150 and K_{G3} at 200. For the same reason as in the SG system it is desirable to lower

the gain slightly. Therefore both K_{G1} and K_{G3} are lowered one step to 100 and 150. This leads to a worse damping ratio but the system is still stable with a reasonable settling time. Fig. 13 shows the response with the lower gain alternative. The system is now stable and all three rotor angles return to the steady state values. The settling time and the least damped mode for both the optimal gain and lower gain alternative is presented in Table VIII. In this case, the lower gain alternatives settling time is slightly longer as expected.

TABLE VIII
STABILIZED GFL SYSTEM COMPARISON

Gain	T_s [s]	Mode [s^{-1}]	f_D [Hz]	ζ_{min} [%]
$K_{G3} = 200$ $K_{G1} = 150$	12.88	$-0.81 \pm j5.66$	0.90	14.10
$K_{G3} = 150$ $K_{G1} = 100$	14.97	$-0.23 \pm j2.41$	0.38	9.52

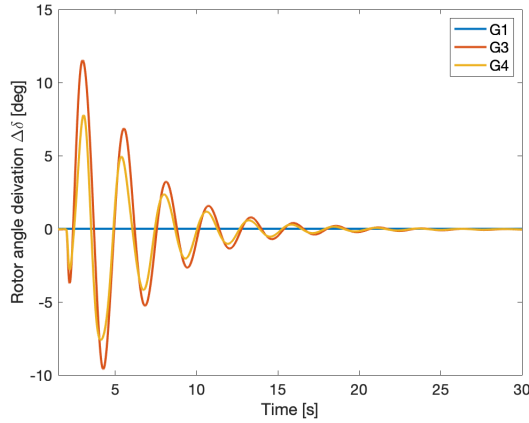


Fig. 13. Rotor angles of G1, G3 and G4 from the fault simulation for the stabilized GFL system with $K_{G3} = 150$ and $K_{G1} = 100$.

E. GFM system

For the GFM system, the wind farm is instead connected to the grid with a grid forming converter emulating a virtual synchronous machine. Fig. 14 shows the rotor angles of G1, G3 and G4 during the fault simulation. Note that this is without any PSS implemented in the system. Despite this the system response is stable with a settling time of 7.87 seconds, which is better than both the SG system and the GFL system, with two PSS implemented. Just like a SG, the GFM provides the grid with a set voltage and frequency reference. However, it does not suffer from the same instability issue caused by the high excitation gain in the SG system. Therefore, the PSSs are not needed in the GFM system. In addition, the power electronics based converter can adapt to the grid conditions faster than the mechanical SG, meaning the power production can be quickly controlled to further improve the response time.

Fig. 15 shows the voltage at terminal two during the fault simulation for the SG system with two PSS, the GFL system with two PSS and the GFM system with no PSS. The voltage is plotted at terminal two since the power source is changed

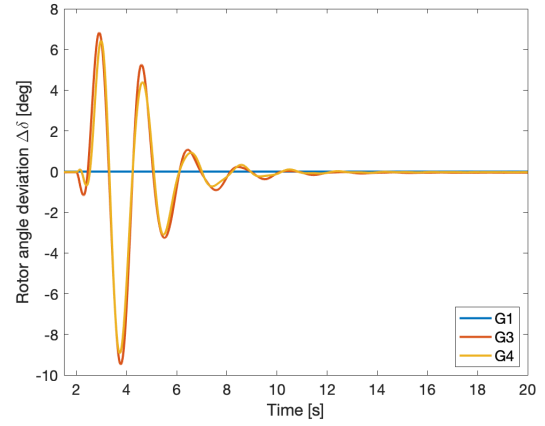


Fig. 14. Rotor angles of G1, G3 and G4 from the fault simulation for the GFM system.

there, and therefore the effects should be most evident there. The GFL system has the largest drop in voltage at the time of the fault and also the largest oscillations. This is to be expected, as it is the only alternative not providing a set voltage reference. Both the SG system and the GFM system have a set voltage reference at terminal two and their voltage oscillations are much smaller, which as previously explained leads to smaller rotor angle oscillations.

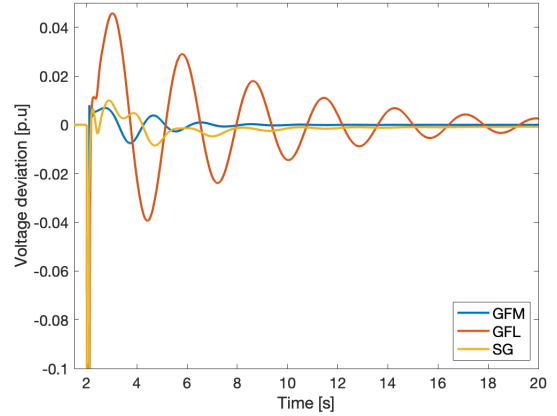


Fig. 15. Voltages at terminal 2 from the fault simulation for the GFM, GFL and SG system.

V. DISCUSSION

A. Effects of inertia

By comparing the SG and GFL system without PSS it is clear that the GFL system's response to the short-circuit fault is less stable. Both systems become small signal unstable after the fault which leads to loss of synchronization between the two areas. However, this occurs faster in the GFL system than in the SG system. As explained in the results, the GFL system's G1 accelerates faster at the loss of active power due to the lower inertia in that side of the system according to Equation 11. In [9] it is described that the GFL can have a positive effect on the system stability assuming there are enough SGs to maintain a minimum level of inertia. However,

if this condition is not met the GFL will have a negative effect on the overall stability of the system. The results of this project indicate that implementing the GFL worsened the overall stability, which could be explained by the minimum level of inertia being reached when the SG is replaced.

B. Stabilizing systems with PSS

The result shows that both the SG and the GFL system can be stabilized by implementing PSS. Both systems have inter-area oscillations before being stabilized. For the SG system, it is possible to achieve a stable system with only one PSS, although an improved settling time is reached with a second PSS. This results in one PSS in each area of the system. The GFL system on the other hand is unstable with only one PSS and requires one PSS in each area to achieve stability. This means that the best stabilization for both systems is achieved when one PSS is placed in each area of the two-area system. However, due to the properties of the GFL system, the response achieved using PSS is still worse for the GFL compared to the SG system.

The method of implementing and tuning the PSS based on modal analysis works sufficiently for stabilizing these systems. However, as discussed in the results there is a problem with having very high gains for the PSS. In this project, only the static gain of the PSS is tuned. It is possible that the high gain could be avoided, and that the small signal stability could be further improved, by also tuning other constants. For example, the lead lag filters in the PSS affects how the system modes are changed by increasing the gain. Tuning the lead lag constants could therefore enable a better system response without the high gain issues. This is an important aspect of the study that could be improved upon.

A limitation of using PSS to stabilize the systems is that this control method can only be applied at the SGs. In a system with low penetration of CBRES this method works sufficiently. However, when the share of CBRES increases the possible placements for a PSS decreases. The loss of inertia also makes it difficult to stabilize the system with PSS, since they do not offer any type of virtual inertia. Therefore PSS are not a complete solution for stabilizing future power systems with higher shares of converter-based power production.

C. The potential of GFM

The SG and the GFL system response indicate that implementing a wind farm with a GFL causes a more vulnerable grid. As discussed, this is due to the loss of inertia which sets a limit for how much CBRES can be implemented. Implementing the wind farm with GFM instead is one possible solution. The results of this project show that the GFM system is more stable than both the GFL and the SG system with the fastest settling time of all the alternatives. By providing a voltage and frequency reference to the grid the GFM can compensate for the lost inertia. Unlike the SG system the GFM does not suffer from the problem of high excitation gain, and the properties of power electronics also allow for a faster response time. This shows that using GFM is a feasible

solution for building stable future power systems with a high penetration of CBRES.

In [9] the prospect of using GFM in power systems is discussed. Research shows that stability can be achieved, even with a large share of CBRES, if roughly 10-30% of the converters in a power system are GFM. Today most converters in commercial use are GFL but there are GFM available that use megawatt batteries as energy reserves. Economic factors are one of the main obstacles for manufacturers of GFM today. Another hindering factor is that there are a lot of options for GFM design and no industry accepted standard, which also contributes to the high cost. Setting a standard for the expected performance of GFM will require more future research and cooperation between actors. Today the performance of GFL is the focus of many manufactures but power systems are reaching the limit of how high CBRES penetration GFL can support. This raises the need for technologies such as GFM, that can contribute to stabilizing future power systems. Therefore the commercial use of GFM will probably increase in the near future. Examining the properties of different GFM designs can be interesting for future projects, since establishing an industry standard is a highly discussed matter today.

VI. CONCLUSION

The results show that PSS can be used to stabilize systems both with and without converter-based power production. Both with and without PSS implemented the GFL system is less stable than the SG system. PSS can be used to improve the small signal stability of power systems with converter-based power production but it is not a complete solution for future systems with a higher penetration of CBRES. On the other hand, implementing power production that instead uses GFM improves the small signal stability. Utilizing GFM is a feasible solution for designing stable power systems with a very high penetration of CBRES. However, for this to become a reality more research and cooperation to establish an accepted standard for the GFM technology is needed.

ACKNOWLEDGMENT

The authors would like to thank our supervisor Angel Clark and Merhadd Ghandhari for their valuable input to the project. We would like to express an additional gratitude to Angel for her expertise, support and commitment that have made this project possible.

REFERENCES

- [1] IPCC, "Summary for policymakers," in *Climate Change 2022: Impacts, Adaptation, and Vulnerability. Contribution of Working Group II to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, Feb 2022.
- [2] OurWorldinData. (2022, Apr) Emissions by sector. [Online]. Available: <https://ourworldindata.org/emissions-by-sector>
- [3] REN21, *Renewables 2021, Global Status Report*. Paris, France: REN21 Secretariat, 2021.
- [4] B. S. Hodge, H. Jain, C. Brancucci, G. Seo, M. Korpås, J. Kiviluoma, H. Holtinen, J. C. Smith, A. Orths, A. Estanqueiro, L. Söder, D. Flynn, T. K. Vrana, R. W. Kenyon, and B. Kroposki, "Addressing technical challenges in 100% variable inverter-based renewable energy power systems," *Wiley interdisciplinary reviews. Energy and environment*, vol. 9, no. 5, 2020.

- [5] F. Milano, F. Dörfler, G. Hug, D. J. Hill, and G. Verbič, “Foundations and challenges of low-inertia systems (invited paper),” in *2018 Power Systems Computation Conference (PSCC)*, 2018, pp. 1–25.
- [6] M. Ghandhari, *Stability of Power Systems an Introduction*. Stockholm, Sweden: Royal Institute of Technology (KTH), 2021.
- [7] R. Rosso, X. Wang, M. Liserre, X. Lu, and S. Engelken, “Grid-forming converters: Control approaches, grid-synchronization, and future trends—a review,” *IEEE Open Journal of Industry Applications*, vol. 2, pp. 93–109, 2021.
- [8] B. Kroposki, B. Johnson, Y. Zhang, V. Gevorgian, P. Denholm, B.-M. Hodge, and B. Hannegan, “Achieving a 100% renewable grid: Operating electric power systems with extremely high levels of variable renewable energy,” *IEEE Power and Energy Magazine*, vol. 15, no. 2, pp. 61–73, 2017.
- [9] J. Matevosyan, B. Badrzadeh, T. Prevost, E. Quitmann, D. Ramasubramanian, H. Urdal, S. Achilles, J. MacDowell, S. H. Huang, V. Vital, J. O’Sullivan, and R. Quint, “Grid-forming inverters: Are they the key for high renewable penetration?” *IEEE Power and Energy Magazine*, vol. 17, no. 6, pp. 89–98, 2019.
- [10] PowerFactory, *Technical Reference DIgSILENT, Grid-forming Converter Templates*. Gomaringen, Germany: DIgSILENT GmbH, 2021.
- [11] S. D’Arco and J. A. Suul, “Equivalence of virtual synchronous machines and frequency-droops for converter-based microgrids,” *IEEE Transactions on Smart Grid*, vol. 5, no. 1, pp. 394–395, 2014.

Assessing the Impact of High Grid Penetration of Renewable Energy on Power System Stability

Alexander Leijonhielm and William Nordberg

Abstract—In this report, the effect that a higher penetration of renewable energy sources has on electric power grid stability is evaluated. The report also compares different methods of stabilizing an unstable grid. The model used is a two-area four-machine system and the main objective is to stabilize the synchronous generators such that they revert back to synchronism after being subjugated to a small signal disturbance. The stabilization methods consists of supplementary Power System Stabilizers (PSSs) complementing the exciter systems of the synchronous machines, as well as two types of converter-based controllers in the renewable energy source: Grid-Following (GFL) converters and Grid-Forming (GFM) converters. The results show that a system with renewable energy sources is more sensitive to disturbances and has a larger rotor angle deviation from a steady state when using only GFLs compared to the conventional grid without PSSs. It is also found that a conventional grid requires supplementary PSSs to be stable. This is also the case for a system with renewable energy controlled by GFL. The system with GFM controllers does however not need supplementary PSS to be stable. This leads to the conclusion that GFM is more preferable than GFL to control a grid with a higher penetration of renewable energy.

Sammanfattning—I denna rapport utvärderas hur en högre andel förnybara energikällor påverkar stabiliteten i elnät, och jämför också olika metoder för att stabilisera ett instabilt nät. Modellen som används var ett två-områdes-fyrmaskinsystem och huvudsyftet är att stabilisera synkrongeneratorerna så att de återgår till synkronism efter att ha utsatts för en liten småsignalstörning. Stabiliseringsmetoderna består av kompletterande Power System Stabilizers (PSS:er) som kompletterade exciteringssystemen i synkronmaskinerna, samt två typer av omvandlarbaserade styrenheter i den förnybara energikällan: Grid-Following (GFL)-omvandlare och Grid-Forming (GFM)-omvandlare. Resultaten visar att ett system med förnybara energikällor är mer känsligt för störningar och har en större rotorvinkelavvikelse från ett stationärt tillstånd när GFL-kontroller används jämfört med det konventionella nätet utan PSS:er. Det visar sig också att ett konventionellt nät kräver kompletterande PSS:er för att vara stabilt. Detta är också fallet för ett system med förnybar energi som enbart kontrolleras av GFL-omvandlare. Systemet med GFM-omvandlare behöver dock inte kompletterande PSS för att vara stabilt. Detta leder till slutsatsen att GFM är mer att föredra än GFL för att kontrollera ett nät med högre andel förnybar energi.

Index Terms—Power System Stability, Variable Renewable Energy, Power System Stabilizers, Grid Following Converters, Grid Forming Converters

Supervisors: Angel Clark, Mehrdad Ghandhari

TRITA number: TRITA-EECS-EX-2022:142

I. INTRODUCTION

Climate change is the most dramatic and ubiquitous crisis humanity has ever encountered. In tandem with industrializa-

tion and electrification, our dependency on fossil fuels has gone up drastically; according to [1], fossil fuels generated almost two-thirds of the world's electricity production in 2015. To combat the longstanding negative effects of this dependency on fossil fuels, new, greener solutions must therefore be implemented in various areas. A number of solutions have been proposed and implemented thus far, namely nuclear fission, hydro power, wind power and solar power. The latter two are so called intermittent sources, meaning they are not continuous and therefore hard to rely on solely. Integrating them without accounting for their drawbacks can lead to a major decrease in power grid stability and therefore cause more power outages as well as higher electricity costs.

Ever since the first commercial generation of electricity in 1882 [2], electric power has almost exclusively been produced by techniques relying on the large inertia of synchronous generators (SGs) to provide stability. However, variable renewable energy (VRE) sources such as wind and solar do not utilize inertia in the same way as they are instead converter-based using Pulse Width Modulation (PWM) and are therefore inherently less stable than SGs. To stabilize current grids with SGs, one can employ additional controllers in the generator exciter systems called Power System Stabilizers (PSS) which uses control theory and negative feedback to adjust the generator's field voltage appropriately. VRE sources are however not compatible with PSSs and therefore require different types of stabilizing methods. The most prevalent controller method today is the Grid-Following Converter (GFL) which follows the grid voltage. The GFL however requires a stable voltage from the grid and will not be sufficient in a grid with a high penetration of VRE sources. According to [3], research is currently being done on so-called Grid-Forming Converters (GFMs) which can improve voltage and frequency stability in a grid by establishing the voltage and frequency themselves without having to use the power grid. All of these methods will be tested and compared in this report.

The aim of this project is to both assess the impact renewable energy has on power system stability, but also test different stabilizing methods for a grid with no penetration of VRE and one with high penetration. To do this, DiGSILENT PowerFactory is used to simulate a two-area four-machine system. The system consists of four generators (of which one is later substituted for a wind farm with 250 wind turbines), eleven buses and two loads. The system model is more thoroughly explained in section II.

II. THEORY

A. Exciter systems

The voltage in the rotor windings in the SG, also referred to as field voltage in [4], is controlled by an exciter system. The basic function of the exciter is to control the terminal 3-phase voltages by varying the field voltage in the rotor. A simplified model of the exciter system of type ESAC4A, which was the one used in the project, can be seen in Fig. 1.

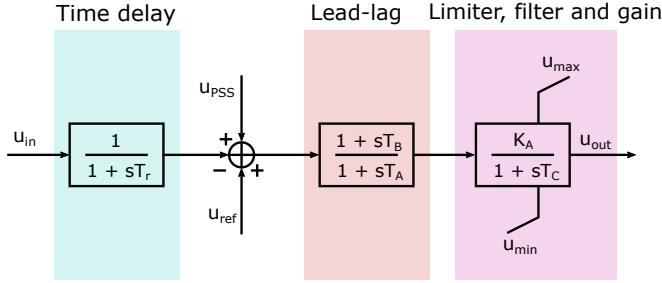


Fig. 1. Simplified model of the ESAC4A exciter system.

The input signal u_{in} is the terminal voltage and the output signal u_{out} is the exerted field voltage in the rotor. The control signal u_{ref} is seen in the addition-block in the diagram. The section highlighted in blue is a time delay, known as a transducer, with time constant T_r . This is used to compensate the time it takes to measure, transform and filter the signal. The other sections in the control block diagram, highlighted in red and purple, is for filtering and stabilizing the output. The red section is a lead-lag filter with lead-constant T_B and lag-constant T_A . The purple section is a low-pass filter with time constant T_C , a limiter, and gain K_A . If a high K_A gain is employed transient stability in the generator will be improved. However, it will also lead to a worse small-signal stability. This is the reason supplementary PSSs are used. The variable u_{PSS} in Fig. 1 represents the PSS's output.

The control block diagram for the supplementary PSS controller is seen in Fig. 2.

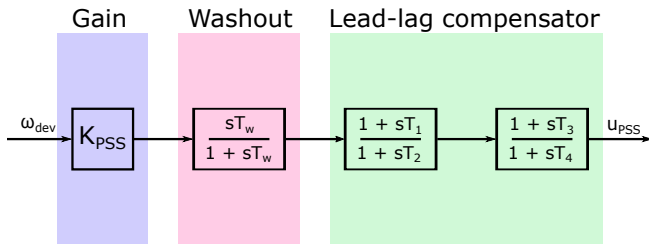


Fig. 2. Control block diagram of a PSS of type STAB1.

The input signal for the PSS is most commonly the generator speed deviation ω_{dev} , i.e. the difference between the current speed and nominal speed.

The first block, highlighted in purple, in the diagram is the stabilizer gain where K_{PSS} is the gain coefficient. This

determines how much small-signal oscillations are dampened at the generator terminal. It is preferable to employ the lowest possible gain that results in good small-signal stability. The second part, highlighted in pink, represents a high pass filter known as a washout. T_w is the time constant of the filter. The purpose of this filter is to mitigate any steady state deviation from the input signal. The third part of the PSS, highlighted in green, is a lead-lag compensator with constants T_1 – T_4 . The constants are tuned so that a negative gain is employed on the oscillations and the root locust of the system becomes stable.

The specific values for all exciter and PSS parameters used in this project can be found in appendix A and B respectively.

B. Power system stability

The stability of a power system refers to the system's ability to return to equilibrium after being subjugated to a disturbance given an initial operating point, which is discussed in detail in [4]. In this project, the type of power system stability investigated was rotor angle stability. This type of stability can be further categorized into small-signal stability and transient stability. Small-signal stability concerns a system's ability to return to a steady state after being exposed to a small-signal disturbance. A disturbance is considered small enough if the system is able to be linearized. Transient stability, however, refers to a system's ability to regain equilibrium after being subjugated to a large disturbance such as a short circuit.

C. Swing equation

The swing equation is very important to understand the dynamic response of a SG and is explained in great detail in [4]. The dynamics can be described by

$$\omega_m J \frac{d\omega_m}{dt} = P_m - P_e \quad (1)$$

with P_m and P_e (in W) as the mechanical and the electrical power respectively and ω_m as the mechanical frequency (in Hz). The inertia constant H is given by

$$H = \frac{W_{Ks}}{S_{ng}} \quad (2)$$

where W_{Ks} is the total kinetic energy stored in the rotor (in J) and S_{ng} is the rated apparent power of the generator (in VAR). H defines how long (in seconds) it takes for the rotor to go from synchronous speed to standstill after a disturbance occurs while still extracting rated power from the generator and not supplying the generator with any mechanical power.

The swing equation (1) can then be written as

$$\dot{\omega} = \frac{1}{M} (P_{mpu} - P_{epu} - D\omega) \quad (3)$$

where P_{mpu} and P_{epu} represent the mechanical and electrical power (in p.u.) respectively. Additionally, the positive constant D is added to represent the damping power, i.e. the impact of the physical behavior of friction in the bearings and other dampening effects. As can clearly be seen, H is the only variable in (3) which means that a lower inertia in a power system will affect its overall stability negatively as the frequency rate of change $\dot{\omega}$ in will increase.

D. Modal analysis

The dynamics of a power system can be modeled by the equations

$$\begin{aligned}\dot{x} &= f(x, y) \\ 0 &= g(x, y)\end{aligned}\quad (4)$$

where $f(x, y)$ and $g(x, y)$ describe the responses of all generators. This non-linear system can be linearized around the steady state point (x_0, y_0) by the following equations:

$$\Delta \dot{x} = f_x \Delta x + f_y \Delta y \quad (5)$$

$$0 = g_x \Delta x + g_y \Delta y \quad (6)$$

where

$$\begin{aligned}f_x &= \left[\frac{\partial f(x, y)}{\partial x} \right]_{x=x_0; y=y_0} & f_y &= \left[\frac{\partial f(x, y)}{\partial y} \right]_{x=x_0; y=y_0} \\ g_x &= \left[\frac{\partial g(x, y)}{\partial x} \right]_{x=x_0; y=y_0} & g_y &= \left[\frac{\partial g(x, y)}{\partial y} \right]_{x=x_0; y=y_0}\end{aligned}$$

are the system's Jacobian matrices. Now, from (6), Δy can be solved as

$$\Delta y = -(g_y)^{-1} g_x \Delta x \quad (7)$$

Substituting this into (5), the following is received

$$\Delta \dot{x} = (f_x - (g_y)^{-1} g_x) \Delta x = A \Delta x \quad (8)$$

where A is the overall system state matrix. In the following parts of the report, whenever an eigenvalue is mentioned it refers to the eigenvalue of the matrix A .

Eigenvalues are often associated with a system's modes. In order for a system to be stable, its modes must be on the left half-plane in the complex domain. Modes on the right half-plane lead to unstable responses. Since eigenvalues are complex quantities, this means that the real part of the mode must be strictly negative for it to be a stable mode. Their complex nature also leads to another important property which is that for each complex mode, its conjugate is also a mode. The i -th conjugate pair is written as

$$\lambda_i = \sigma_i \pm j\omega_{p_i} \quad (9)$$

where σ_i is the real component and therefore dictates whether the mode is stable or unstable. The imaginary component ω_{p_i} gives the oscillation frequency of the mode. It's expressed as

$$\omega_{p_i} = 2\pi f_{p_i}. \quad (10)$$

The damping ratio of a given mode is given by

$$\zeta_i = \frac{-\sigma_i}{\sqrt{\sigma_i^2 + \omega_{p_i}^2}} \quad (11)$$

and dictates whether the response of a system will diverge or converge to 0 after a given amount of time.

The right eigenvector V_i^r corresponding to mode i is given by any non-zero vector V_i^r which satisfies

$$A V_i^r = \lambda_i V_i^r. \quad (12)$$

In the same way, the left eigenvector is given by any non-zero vector V_i^l which satisfies

$$V_i^l A = V_i^l \lambda_i \quad (13)$$

Using these right and left eigenvectors, it is convenient to introduce the following modal matrix:

$$\begin{aligned}V^R &= [V_1^r \ V_2^r \ V_3^r \ \cdots \ V_{n_x}^r] \\ &= \begin{bmatrix} v_{11}^r & v_{12}^r & \cdots & v_{1n_x}^r \\ v_{21}^r & v_{22}^r & \cdots & v_{2n_x}^r \\ \vdots & \vdots & \ddots & \vdots \\ v_{n_x1}^r & v_{n_x2}^r & \cdots & v_{n_xn_x}^r \end{bmatrix}. \end{aligned} \quad (14)$$

In control problems, it is of great importance to know which state variables mostly affect the dynamic of a given mode. However, due to the state matrix A seldom being a diagonal matrix, the state variables are often linear combinations of other state variables and therefore it's difficult to identify which state variables to regulate in order to achieve the desired result. To help with this, we introduce the participation factor

$$p_{ki} = v_{ik}^l v_{ki}^r = v_{ki}^r v_{ik}^l \quad (15)$$

which is a measure of the relative participation of the k -th state variable in the dynamic of the i -th mode.

E. Converter-Based Energy Sources and Virtual Synchronous Machines

Apart from the time variance that is inherent to VRE sources, another aspect of importance is their interface to the main power grid. As has been established extensively previously in this report, conventional energy sources interface the main grid with synchronous machines with large rotating masses and mechanical inertia. VRE sources, however, do not have any rotating masses at all since they interface the grid with electronic converters and PWM instead of synchronous machines.

It might at first seem counter-intuitive to state that all VRE sources lack mechanical inertia. It is obvious to understand that this is true for photovoltaic sources, but when it comes to wind power it might seem contradictory as wind generators have huge rotating rotor blades to turn a turbine. The reason as to why this does not contribute to the overall inertia in an interconnected system, as is explained in [5], is that the natural frequency with which a wind turbine turns is intermittent. This would not be compatible with the main grid and its nominal frequency of often 50 Hz or 60 Hz. The way to work around this problem is to use voltage rectifiers and then create a sinusoidal voltage with nominal frequency using electronic converters and PWM. This means that even if wind turbines have huge rotating bodies, they contribute no inertia to the system.

All of this means that a power system with a large amount of VRE sources will have a lower inertia M . Consulting the swing equation (3), it is easy to see that this will result in the speed of the synchronous machines left in the system will be more vulnerable to disturbances. Also due to the lack of synchronous generators, the stabilizing methods previously

discussed in this report (i.e. exciter systems with PSSs) is rendered unusable.

Due to the lack of mechanical properties in converter-based technologies, VRE interfaces rely on other types of control algorithms instead. According to [5], there are mainly two classes of converter controllers: Grid Following (GFL) and Grid Forming (GFM). Today, the most prevalent of these is the GFL controller which can be modeled as a current source I_C with a high parallel impedance Z_{GFL} . The GFM controller on the other hand can be modeled as a voltage source E_C with a low series impedance Z_{GFM} . The terminal voltage is labeled V_C in fig. 3 and 4. The rest of the grid is modeled as a voltage source V_G with a series impedance Z_G . 3 shows conceptual models of the two control approaches whilst fig 4 shows the responses of the control models when subjugated to a disturbance.

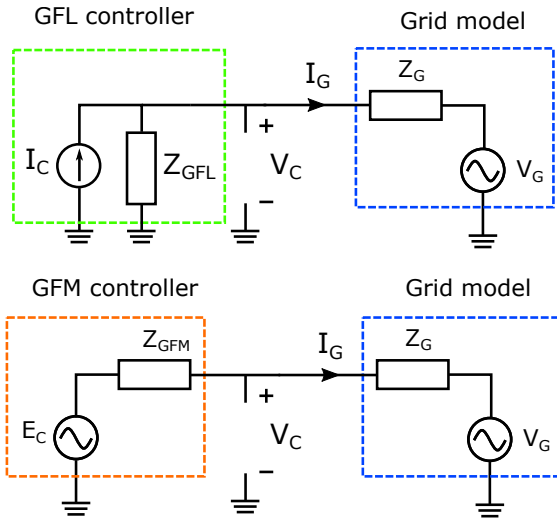


Fig. 3. Conceptual models of GFL and GFM controllers connected to a grid.

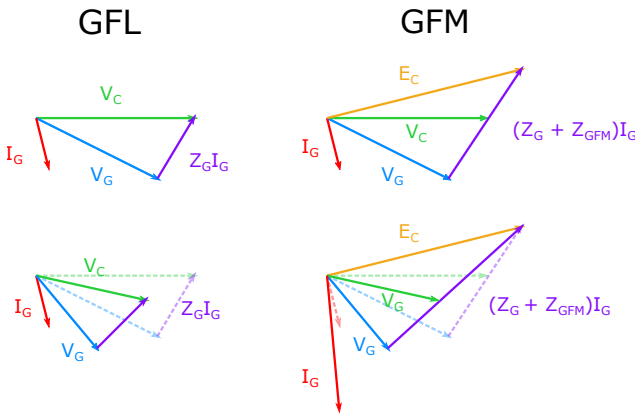


Fig. 4. Controller phasors after being subjugated to a disturbance, i.e. a change in V_G .

The GFL converters synchronize their outputs with the main grid by continuously estimating the phase angle of the

terminal voltage with a Phase Locked Loop (PLL). These modeling details are presented thoroughly in [6]. Conceptually, this means that GFL controllers mimic the terminal voltage. GFL converters do however have a few problems. Due to their intrinsic nature as a current source, they will keep the I_C phasor steady while the V_C phasor may vary more under a disturbance or a changing V_G phasor, which is not a desired quality as a stiff voltage is of great importance in a stable grid. Another problem with GFL controllers is that they only work under the assumption that the grid has a large portion of SGs, or at least sources capable of providing a stiff voltage themselves. Since it may be desired to at some point completely replace all SG energy sources dependent on fossil fuels in a power grid with converter based sources, GFL controllers will not be sufficient.

In [5] and [7] it is proposed that GFM controllers do not have these limitations. GFMs do not require a stiff voltage to operate because they are modeled as a sinusoidal voltage source by emulating a synchronous machine in software. This is referred to as a Virtual Synchronous Machine (VSM) which inherits virtual inertia. According to [8], virtual inertia refers to how the phase angle of the complex terminal voltage varies as it would in a real synchronous machine, i.e. the VSM calculates the phase θ_{VSM} in fig. 5. As mentioned in [9], the most simple way to simulate a synchronous machine is by using the swing equation (3). A control block diagram explaining the emulations of a synchronous machine based on the swing equation is shown in Fig. 5.

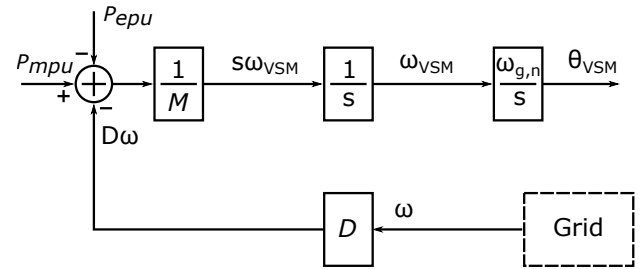


Fig. 5. Control block diagram of a VSM based on the swing equation.

However, it should be noted that due to the GFM's voltage source characteristics, the current through the power electronics implementing the controller may vary drastically under short periods of time. This can be seen in Fig. 4 where the I_g phasor shifts in the GFM. As mentioned in [6], proper fault current detection techniques must be implemented to mitigate any damage done to the power electronics.

F. System model

The system model is based on the widely used Kundur two-area system, which is explained thoroughly in [10]. The system has a frequency of 50 Hz. The base system consists of eleven buses, four generators (two per area) each with a rated apparent power of 900 MVA and the power transfer between the areas was 400 MVA. Each generator is accompanied by a step-up

transformer with a voltage ratio of 20 kV/230 kV. The power flow in the generators in the base model can be seen in Table I and the power flow in the system with wind can be seen in Table II. Fig. 6 shows the conceptual structure of the two-area system.

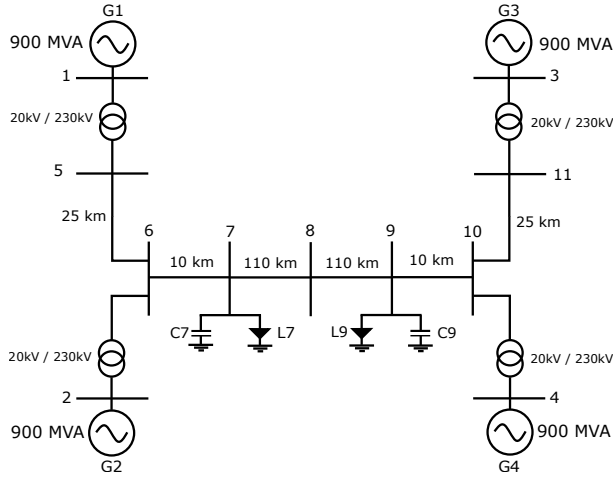


Fig. 6. The Kundur two-area system model.

TABLE I
POWER FLOW OF SYSTEM AT OPERATING POINT WITHOUT WIND POWER

Generator	P (MW)	Q (MVar)
G1	697	175
G2	700	212
G3	719	109
G4	700	159

TABLE II
POWER FLOW OF SYSTEM AT OPERATING POINT WITH WIND POWER

Generator	P (MW)	Q (MVar)
G1	698	175
G2	700	212
G3	719	109
Wind Farm	700	189

III. METHODOLOGY

To measure stability and response time, a disturbance had to be introduced. This was done by simulating a 100 ms solid short-circuit fault between bus 7 and 8 at $t = 2$ s after simulation start. Since the duration of the fault was so short on such a long transmission line, it was sufficiently small to be considered a small-signal disturbance. This means modal analysis was a viable method to evaluate the system.

Before conducting any simulations, it was necessary to determine which results would be deemed satisfactory. The main objective was to stabilize the grid. Therefore, the grid was considered stable if a settling time of roughly 10 seconds for the rotor angle deviation was achieved. This meant that if one PSS did not achieve the desired settling time, another

one or two supplementary PSSs would be installed to improve stability.

A. Identification of the critical generator

In order to stabilize the grid, it was necessary to determine which generator was causing the most stability issues. Hence, a load flow analysis was performed to reset the operating point. Thereafter an eigenvalue analysis was performed to find the most unstable mode. The participation factor for this particular mode was then calculated to find which state variable (and in effect which generator) was most prominent in the destabilization of the mode.

B. PSS installation

After the critical generator had been identified, a PSS of type STAB1 was installed at said generator. In order to find the optimal controller gain for the PSS, the gain was increased in increments of 10; for each increment, modal analysis was performed and the damping ratio of the most unstable mode was noted. If the change in damping ratio with respect to the increased gain did not converge, the gain was capped at 130 since it's preferable to not have a gain much higher than 150 (for reasons explained in section II-A) and therefore having the gain set at 130 left some margin.

If a settling time of roughly 10 seconds could not be achieved with a single PSS at a satisfactory gain, the process was repeated from section III-A again with another PSS. The second PSS was then installed in the generator with the highest participation factor for the system with one PSS.

C. Integration of wind power

The generator G4 in Fig. 6 was then replaced with a wind farm of 250 wind turbines connected to the grid via 250 transformers in bus 4. This means that the wind power was installed on the receiving end of the system. Each wind turbine had a rated power of 3.6 MVA, resulting in a power output of roughly 900 MVA, the same as generator G4 had previously been supplying the grid with.

The method of stabilizing the new system began by only using a GFL controller. If this was not deemed sufficient, PSSs would be installed in the remaining SGs in accordance of the process described in sections III-A–III-B.

The system was then instead stabilized using only GFM controls in the wind power source directly.

IV. RESULTS

Fig. 8 shows the system's dynamic and the response to a 100 ms fault at bus 8 with no wind power integrated. It is evident due to the converging nature of the graphs that the system is unstable and needs stabilizing. A PSS was therefore installed at the critical generator G3 at bus 3. The change in damping factor with respect to increased gain can be seen in Fig. 9. As seen in the graph, the damping factor did not converge for a low enough gain. Therefore, the gain K_{G3} was set at 130. This did not result in a good enough response however and another PSS was therefore installed in the second-most critical

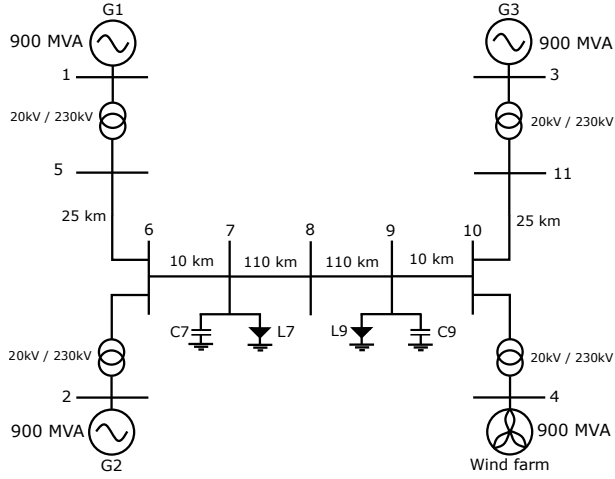
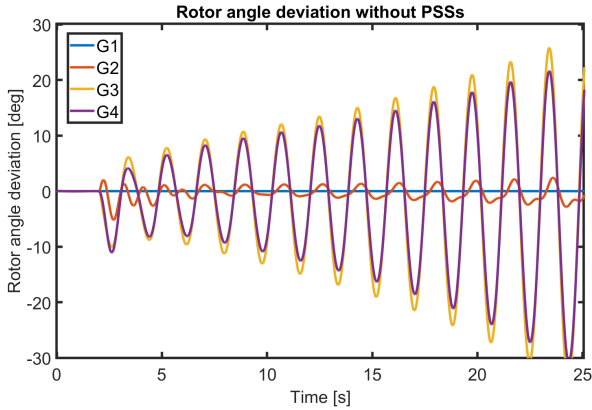


Fig. 7. System model with a wind farm at terminal 4.

generator G2. The tuning of the gain for this PSS yielded the same non-converging result as the previous PSS and the gain K_{G2} was therefore set at 130 as well. This produced the much better response shown in Fig. 10. This led to a settling time of 9.36 seconds. Table III shows the least stable modes and their frequencies for zero, one and two PSSs installed respectively. The frequencies of the modes were also calculated by hand with (10) and did not differ from the simulated values except for ω_{p5} which was 0.02 Hz lower than calculated. Evidently, the simulation yields satisfactory results.

Fig. 8. Rotor angle deviation as the system without any PSSs is subjected to a 100 ms fault at $t = 2$ s. The rotor angle deviation of each generator is with respect to its base angle as opposed to showing the deviation from the reference machine G1's angle.TABLE III
MODAL DATA FOR CONVENTIONAL SYSTEM

Number of PSSs	Gain	Least stable mode	Damping ratio (%)	Frequency (Hz)
0	–	1	-3.47	0.55
1	130	4	3.88	0.47
2	130	5	12.56	0.44

The initial response of the system with wind power installed

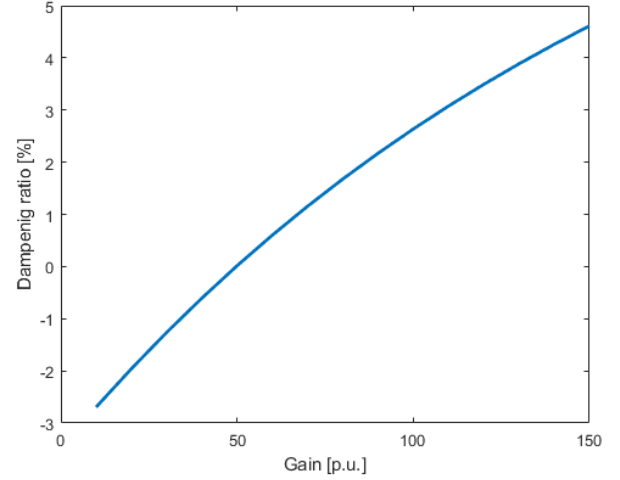
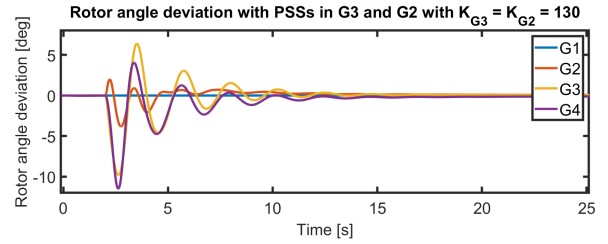
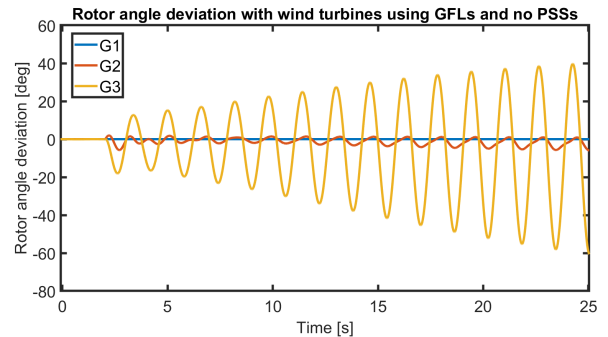


Fig. 9. Damping ratio plotted against PSS gain.

Fig. 10. Rotor angle deviation as the system with two PSSs in G3 and G2 is subjected to a 100 ms fault at $t = 2$ s.

and no PSSs is shown in Fig. 11. It is clear that the largest oscillations of this system have much larger amplitudes than the ones in the base system shown in Fig. 8. Eigenvalue analysis identified the critical generator as generator G3 and a PSS was therefore installed. The gain converged at $K_{G3} = 70$. This did not yield a good enough settling time and therefore another PSS was installed in the second-most critical generator G2. The gain K_{G2} did not converge at a low enough level and was therefore set to 130. The final response is shown in Fig. 12. As can be seen, this resulted in a settling time of 6.77 seconds.

Fig. 11. Rotor angle deviation as the wind power system with GFL and no PSSs is subjected to a 100 ms fault at $t = 2$ s.

The PSSs in the VRE system was then disengaged and the

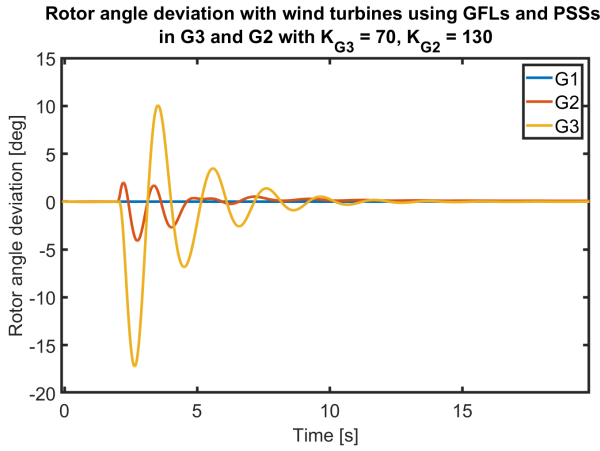


Fig. 12. Rotor angle deviation as the wind power system with GFL and two PSSs is subjected to a 100 ms fault at $t = 2$ s.

system was instead stabilized with the help of GFM converters. The response can be seen in Fig. 13. A slightly quicker settling time of 5.24 seconds was achieved with this method. However, the GFM-based stabilized system has other preferable characteristics as opposed to the GFL-based stabilized system. Fig. 14 shows the difference in internal frequency and internal voltage in the GFL system (complemented by PSSs in generator 2 and 3) and the GFM system. Both the frequency and the voltage stability of the GFM is significantly better, even though the GFL was supported by PSSs in the system.

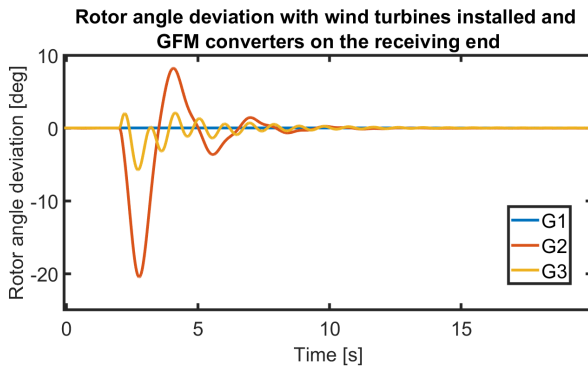


Fig. 13. Rotor angle deviation as the wind power system with a GFM is subjected to a 100 ms fault at $t = 2$ s.

V. CONCLUSION

The simulations of this project led to the conclusion that a conventional system with only synchronous generators required supplementary PSSs to be stabilized. This was also the case for a grid with integrated wind power which utilized GFL. The system with GFM did not however require any supplementary PSS to be stable. It also upheld a much better internal frequency and voltage during the disturbance making it a more preferable choice of controlling wind power in the context that was analyzed.

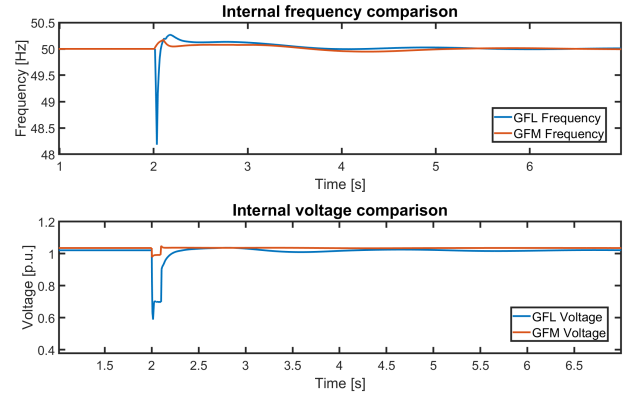


Fig. 14. Internal frequency and voltage of the GFL and GFM controllers. The GFL was supported by PSSs in the system during this measurement.

VI. DISCUSSION

The results of this project show that electrical power grids can be sufficiently stabilized by the use of one or more PSSs after a disturbance have taken place, given that the amount of VRE penetration is low. This can clearly be seen in Fig. 10 where the rotor angle deviation returns to a steady point within roughly 10 seconds.

However, it is evident that while a higher penetration of VRE can reach that same stability with the use of two PSSs (as shown in Fig. 12), it is not optimal to use this method since each replaced SG also removes a place for the PSS to reside. It is also clear from Fig. 11 that the degree of instability of the grid with the added wind farm was significantly higher than the one without the wind farm with a larger max rotor angle deviation of a factor $\frac{\Delta\delta_{wind}}{\Delta\delta_{SG}} \approx 2.48$. This is most likely due to the loss of mechanical inertia in the grid as a result of fewer SGs. The instability would also therefore increase if more SGs were replaced with VRE sources. As this is desirable to phase out fossil energy, this is an important conclusion.

Fig. 11 shows that the GFL controller is not capable of stabilizing the grid by itself without supplementary PSSs. However, the GFM controller severely outperformed the GFL equivalent in nearly every way as can be seen in Figs. 13 and 14. It was sufficient to stabilize the grid by itself and also the fastest stabilizing method tested in this report. Although it does have a larger maximal deviation from synchronism initially after the disturbance. Since GFMs are still in their infancy, it is possible that more sophisticated versions later on mitigate this problem.

A way to expand on this report in the future would be to compare different kinds of GFM controllers instead of comparing them with other types of controls as has been done in this project. It would also be of interest to analyze if the results would be the same if the system was subjugated to different types of disturbances and also a larger amount of integrated wind power.

APPENDIX A EXCITER VALUES

TABLE IV
EXCITER VALUES

Symbol	Quantity	Value
T_r	Measurement Delay [s]	0.01
T_b	Filter Delay Time [s]	0.05
T_c	Filter Derivative Time Constant [s]	0.05
K_a	Controller Gain [p.u.]	200
T_a	Controller Time Constant [s]	0.01
V_{max}	Controller Maximum Output [p.u.]	20
K_c	Rectifier Regulation Constant [p.u.]	0.1
V_{imin}	Input Signal Minimum Limiter [p.u.]	-2
V_{rmin}	Controller Minimum Output [p.u.]	-20
V_{imax}	Input Signal Maximum Limiter [p.u.]	2

APPENDIX B PSS VALUES

TABLE V
PSS VALUES

Symbol	Quantity	Value
K_{PSS}	Stabilizer Gain [p.u.]	varies
T_w	Washout integrate time constant [s]	10
T_2	Second Lead/Lag derivative time constant [s]	0.5
T_4	Second Lead/Lag delay time constant [s]	0.05
T_1	First Lead/Lag derivative time constant [s]	0.5
T_3	First Lead/Lag delay time constant [s]	0.05
$HLIM$	Signal PSS maximum [p.u.]	0.03

ACKNOWLEDGMENT

The authors would like to thank their supervisor Angel Clark who supervised throughout the entire project and gave great advice, literature and insights to the project group.

REFERENCES

- [1] V. Smil, *Energy Transitions: Global and National Perspectives, 2nd Edition*, 2nd ed. Santa Barbara, United States of America: Praeger, Dec. 2016, p. 50.
- [2] —, *Numbers Don't Lie: 71 Things You Need to Know About the World*. Harlow, England: Penguin Books, Apr. 2021, p. 187.
- [3] M. Chen, D. Zhou, A. Tayyebi, E. Prieto-Araujo, F. Dorfler, and F. Blaabjerg, "Generalized multivariable grid-forming control design for power converters," *IEEE Transactions on Smart Grid*, 2022. [Online]. Available: <https://doi.org/10.1109/tsg.2022.3161608>
- [4] M. Ghandhari, "Stability of Power Systems - An introduction," 2021.
- [5] B. Kroposki, B. Johnson, Y. Zhang, V. Gevorgian, P. Denholm, B.-M. Hodge, and B. Hannegan, "Achieving a 100% renewable grid: Operating electric power systems with extremely high levels of variable renewable energy," *IEEE Power and Energy Magazine*, vol. 15, no. 2, pp. 61–73, 2017.
- [6] R. Rosso, X. Wang, M. Liserre, X. Lu, and S. Engelken, "Grid-forming converters: Control approaches, grid-synchronization, and future trends—a review," *IEEE Open Journal of Industry Applications*, vol. 2, pp. 93–109, 2021.
- [7] J. Matevosyan, B. Badrzadeh, T. Prevost, E. Quitmann, D. Ramasubramanian, H. Urdal, S. Achilles, J. MacDowell, S. H. Huang, V. Vital, J. O'Sullivan, and R. Quint, "Grid-forming inverters: Are they the key for high renewable penetration?" *IEEE Power and Energy Magazine*, vol. 17, no. 6, pp. 89–98, 2019.
- [8] DiGSILENT, "Droop controlled converter, synchronverter, virtual synchronous machine," *DiGSILENT Grid-forming Converter Templates*.
- [9] S. D'Arco and J. A. Suul, "Equivalence of virtual synchronous machines and frequency-droops for converter-based microgrids," *IEEE Transactions on Smart Grid*, vol. 5, no. 1, pp. 394–395, 2014.
- [10] R. Varma, R. Mathur, G. Rogers, and P. Kundur, "Modeling effects of system frequency variation in long-term stability studies," *IEEE Transactions on Power Systems*, vol. 11, no. 2, pp. 827–832, 1996.

Design of a Future Residential DC Microgrid

Max Gugolz and Viktor Andersson

Abstract—In the search for environmentally friendly methods to implement renewable energy in the power system residential microgrids have been proposed and proven. The direct current (DC) microgrid topology is a promising implementation of a microgrid system due to the increasing amount of DC-operated loads and production units expected in the near future. In this project, the proposed DC microgrid consists of a solar photovoltaic (PV) power source, a battery, a DC load, and an interlinking bidirectional converter to connect the microgrid to the external three-phase power grid. The PV system is controlled with a Maximum Power Point Tracking (MPPT) algorithm to maximise the power production in all weathers. The DC bus voltage is stabilised by the battery controller and a coordinated control scheme considering the electricity price and battery State of Charge (SOC) is implemented to govern the power exchange with the utility grid. Simulations of the system are shown to validate the functionality of the microgrid and the performance of the controllers in multiple scenarios. The proposed DC microgrid is proven to function in both utility grid-connected mode and in isolation from the utility grid.

Sammanfattning—I sökandet efter miljövänliga metoder att implementera förnybar energi i kraftsystemet har lokalt självförsörjande elsystem för bostäder föreslagits och visats fungera. Den likströmsbaserade topologin är en lovande implementering av ett sådant lokalt elsystem till följd av den ökande mängden likströmsdrivna laster och produktionsenheter som förväntas komma inom en snar framtid. I detta projekt består det föreslagna likströmsbaserade elsystemet av en solenergikälla, ett batteri, en likströmslast och en sammanlänkande dubbelriktad omvandlare för att ansluta det lokala elsystemet till det externa trefasiga elnätet. Solenergisystemet styrs med en maximal kraftpunktföljande algoritm för att maximera kraftproduktionen i alla väder. Likströmsbussens spänning stabiliseras av batteristyrheten och ett samordnat styrschema som tar hänsyn till elpriset och batteriets laddningstillstånd implementeras för att styra energitrycket med elnätet. Simuleringar av systemet presenteras för att validera mikronätets funktionalitet och styrteknikens prestanda i flera olika scenarier. Det föreslagna likströmsbaserade lokala elsystemet visas fungera i både nätanslutet läge och isolerat från elnätet.

Index Terms—Grid control, grid converter, DC microgrid, buck-boost, PV, MPPT, P&O, ESS, SOC.

Supervisors: Qianwen Xu and Mengfan Zhang

TRITA number: TRITA-EECS-EX-2022:143

I. INTRODUCTION

For a long time, conventional fossil-fuelled power plants have been used to produce power and electricity, resulting in large amounts of greenhouse gas emissions. This is warming up the planet and causing changes in the earth's climate. To prevent this, the Paris Agreement was signed in 2015 with a goal to limit global warming well below 2 °C, preferable below 1.5 °C, compared to temperatures before the Industrial Revolution. In 2021 the Glasgow Climate Pack was signed and

this report further points out what has to be done to reach the goals of the Paris Agreement. It is of the utmost importance that all fossil fuel-driven power supplies are replaced with Renewable Energy Sources (RES) if we want to reach the goals [1], [2].

Microgrids are perfectly aligned with the shift toward a higher penetration of RES as they simplify the implementation of PV power sources and wind power systems as well as Energy-Storage Systems (ESS) close to the consumption. Other benefits are improved power quality, reliability and efficiency due to its isolated operation capabilities and controllability. Today most power systems worldwide are alternating current (AC) based but due to increasing amounts of DC loads, batteries and the DC nature of most RES, high interest in DC microgrids has emerged. In the future more electrical vehicles are expected, leading to even more DC devices (batteries) connected to residential homes. With a high rate of DC-based devices in the system, a DC-based grid can avoid unnecessary AC-DC and DC-AC conversions. DC distribution is also more efficient than AC as a result of having no skin effect in the conductors and also no reactive power in the system [3]–[5]. In this paper, one implementation of a DC microgrid is presented and the different parts are described. The microgrid can operate both in connection with the utility grid and in isolation from it. The results are verified through simulations in Simulink.

II. SYSTEM MODEL

The system model simulated in Simulink forms a DC microgrid consisting of a PV power source, a lithium-ion battery, a variable DC load around 800 W, and the power electronics required to connect the subsystems to the DC bus and the bus to the main grid. Consequently, a total of three power electronic converters are in service. A boost converter with MPPT for the PV system, a bidirectional buck-boost converter to enable battery control of the DC bus and finally an interlinking bidirectional converter as a connection between the DC bus and the utility grid. The utility grid is modelled as a strong three-phase AC grid. A conceptual representation of the DC microgrid topology is illustrated in Figure 1. Since the microgrid is utilising an ESS it can operate in both grid-connected and islanded conditions. In other words, the microgrid is designed to function for a limited time even if the utility grid experiences a blackout and is disconnected from the system. In the following subsections, each subsystem will be presented in detail.

A. Battery system

In order to sustain a stable DC bus voltage in the microgrid, both when the utility grid is connected and disconnected to

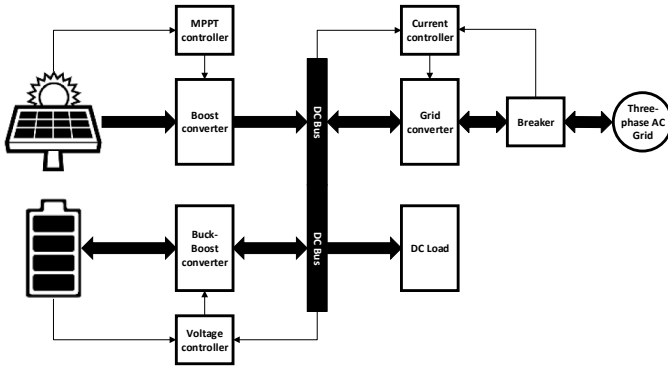


Fig. 1. Topology of the DC microgrid. Thick arrows represent power flow with allowed direction. Thin arrow represent measured signals.

the system, a properly controlled bidirectional buck-boost converter is required for the battery. The controller implemented in this model operates on the measured DC bus voltage and the battery current to sustain a DC bus voltage on a level according to the reference value set to 800 V. The control architecture is illustrated in Figure 2 and it is a cascade control arrangement with an inner current loop and an outer voltage loop to achieve the specified functionality. It is important that the bandwidth of the Pulse-Width-Modulation (PWM) generator is an order of magnitude larger than the bandwidth of the current loop, and that the current control loop is an order of magnitude larger than the voltage control loop, to establish an effective cascade control with satisfying dynamics and dampening [6].

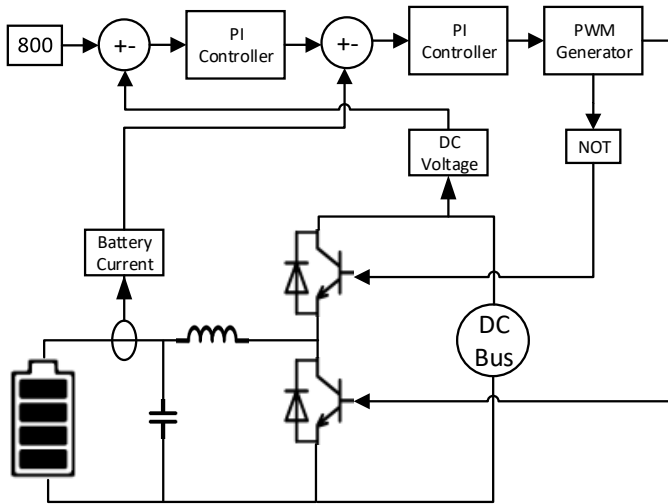


Fig. 2. The battery converter system with its control architecture.

The buck-boost converter structure is realised with two Insulated-Gate Bipolar Transistors (IGBT) and filters to reduce oscillations originating from the switching of the transistors. When the battery is providing power to the DC bus the converter is operating with the boost IGBT and the anti-parallel diode of the buck IGBT as the DC bus voltage of 800 V is higher than the battery nominal voltage of 360 V. However, when the battery is drawing power from the DC bus the opposite IGBT actions play out, with the buck IGBT and the anti-parallel diode of the boost IGBT in operation [7].

B. Photovoltaic system

As the PV module is connected to the load it does not automatically operate at the voltage point corresponding to maximum power production since it is unregulated, and therefore efficiency is lost. Additionally, the optimal operating voltage will shift with a fluctuating irradiance and temperature. The irradiance dependency of the Maximum Power Point (MPP) is illustrated in Figure 3. To solve this, an MPPT algorithm is implemented to control the boost converter built with one IGBT, one diode and filters. This converter enables a controllable voltage at the PV end of the converter with a constant DC bus voltage at the other end.

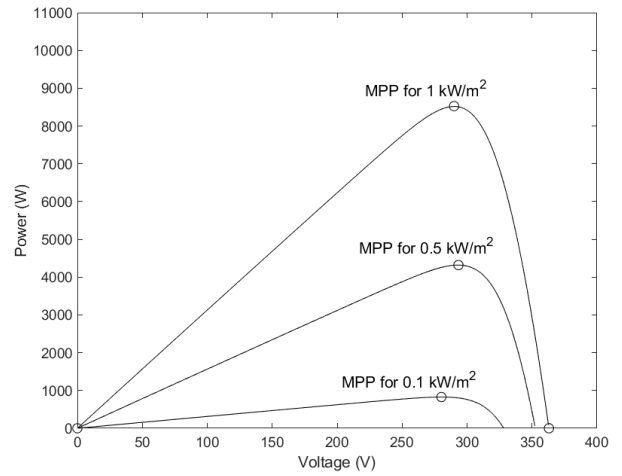


Fig. 3. The MPP is at a different voltages for different irradiances.

The version of the MPPT algorithm used in this system is called Perturb and Observe (P&O) and it is the most powerful method to obtain maximum power from PV arrays [8]. Voltage and current measurements from the PV array are inputted to the P&O algorithm, and after logical and numerical calculations, a reference voltage is outputted and transformed with a Proportional Integral (PI) controller and PWM generator to the duty ratio that controls the boost converter. Figure 4 illustrates the control architecture.

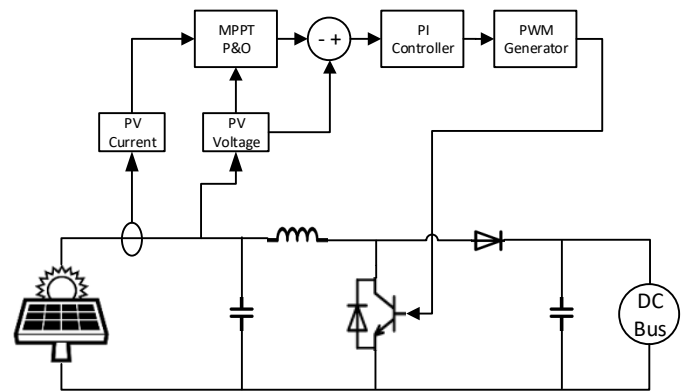


Fig. 4. The PV converter system with its MPPT control architecture.

From the measured PV voltage and current, the P&O algorithm calculates the power and compares it with the power

calculated in the previous iteration to determine if the previous voltage perturbation provided an increase or decrease in observed power. If the power increases, the reference voltage will keep adjusting in the same direction as in the previous iteration. If the power decreases, the reference voltage will be incrementally adjusted in the other direction to dynamically aim for the maximum power point [9]. A flow chart of the P&O algorithm can be seen in Figure 5.

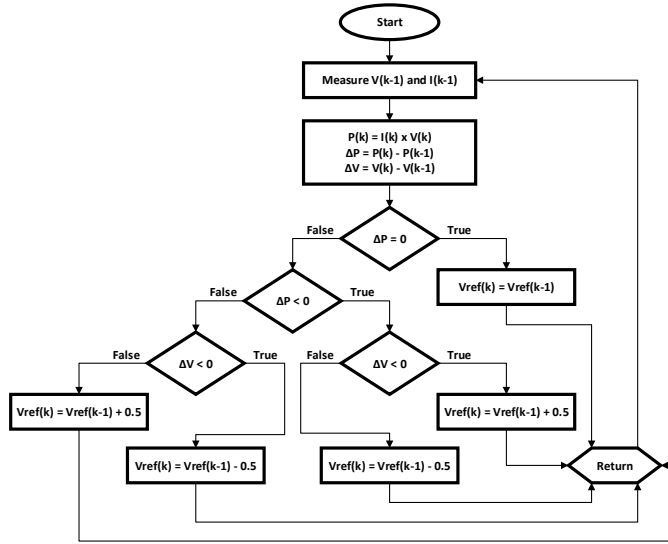


Fig. 5. Flowchart for the P&O algorithm.

C. Grid Converter system

The main functionality of the three-phase bidirectional grid-connected converter is to enable power transfer to and from the utility grid when electricity is bought or sold by the microgrid. Thus, this converter is controlled with a reference current provided by the coordinated control for the microgrid. The converter is implemented using six IGBTs as switches in a bridge formation, with a three-phase LCL filter to reduce harmonics on the current delivered to the main grid.

The corresponding controller that is providing the PWM signal to the converter is illustrated in Fig 6. It is a synchronous dq reference frame controller requiring Park's transformation to be performed on the measured voltages and currents before they are fed into the controller. Park's transformation transforms the signals into a rotating reference frame that rotates synchronously with the utility grid voltages. This produces DC signals that are easily filtered and controlled [10]. The top branch in Figure 6 controls the active current fed to or drawn from the main grid. Thus, this is where the reference current from the coordinated control is inputted. The bottom branch in the same figure controls the reactive current and since no reactive power is preferably produced in these simulations, the reactive current reference is set to zero. To improve the PI controller performance, cross-coupling terms and voltage feed-forward are implemented in the architecture according to [7], [11]. This can be seen in Figure 6 to the right of the PI controllers in both branches. The last steps in the controller consist of transforming the dq signals back to the ordinary abc

reference frame and feeding the reference signal into the PWM generator to compute the pulses used to control the IGBTs.

Synchronisation with the main grid voltage is achieved with a phase-locked loop (PLL) that is dynamically extracting the angle ω_t of the grid voltage and delivering it to the controller. Other ways to extract ω_t from the utility grid voltage are viable but the PLL technique is common in distributed power generation systems [12], [13]. The PLL is operating on the $\alpha\beta$ reference frame achieved by performing Clarke's transformation on the measured grid voltages. If the three phases are balanced then this mathematical transformation results in only two voltages, α and β , instead of the three voltages abc in the regular coordinate system [14].

III. COORDINATED CONTROL

A coordinated control is implemented to control the power exchange between the utility grid and the microgrid. With a coordinated control scheme, the microgrid can select to either buy or sell power to the grid depending on a number of parameters. In this case, the governing parameters are the SOC of the battery and the price of the power at the utility grid. The output of the coordinated control is the reference current going into the controller for the bidirectional grid-connected converter and both the amplitude and the direction of the current are specified. To buy power from the utility grid, often leading to the battery being charged, the reference current is set to a positive value so the flow of power goes from the utility grid to the microgrid. To sell power to the utility grid the opposite is implemented, a negative value for the reference current so the flow of power is in the direction from the microgrid to the utility grid. In order to maintain a power reserve if the utility grid is disconnected, the SOC is preferred to be over 20 %. The coordinated control can be described by a flow chart, as in Figure 7, and is implemented with code in Matlab. The controller is only operating when the microgrid is connected with the utility grid since in isolated mode there can be no power flow to or from the utility grid. When the grid is connected, the coordinated control first inspects if the SOC is over 20 %. The battery is charged if the SOC is lower than 20 %. Otherwise, the electricity price at the utility grid is the decisive parameter. If the price is lower than a selectable value the microgrid is buying power from the utility grid and the other way around if the price is higher than the set value. In order to not damage the battery, the coordinated controller will always sell power if the SOC is over 80 % regardless of the price of the power at the utility grid.

IV. SIMULATION RESULTS

The results from simulations of the DC microgrid are split into three subsections corresponding to the three different simulations performed. The first one focuses on the transitions between utility grid-connected and islanded operations. The second one simulates different electricity prices and the last one simulates islanded operation with varying irradiance on the PV system and varying load levels on the microgrid.

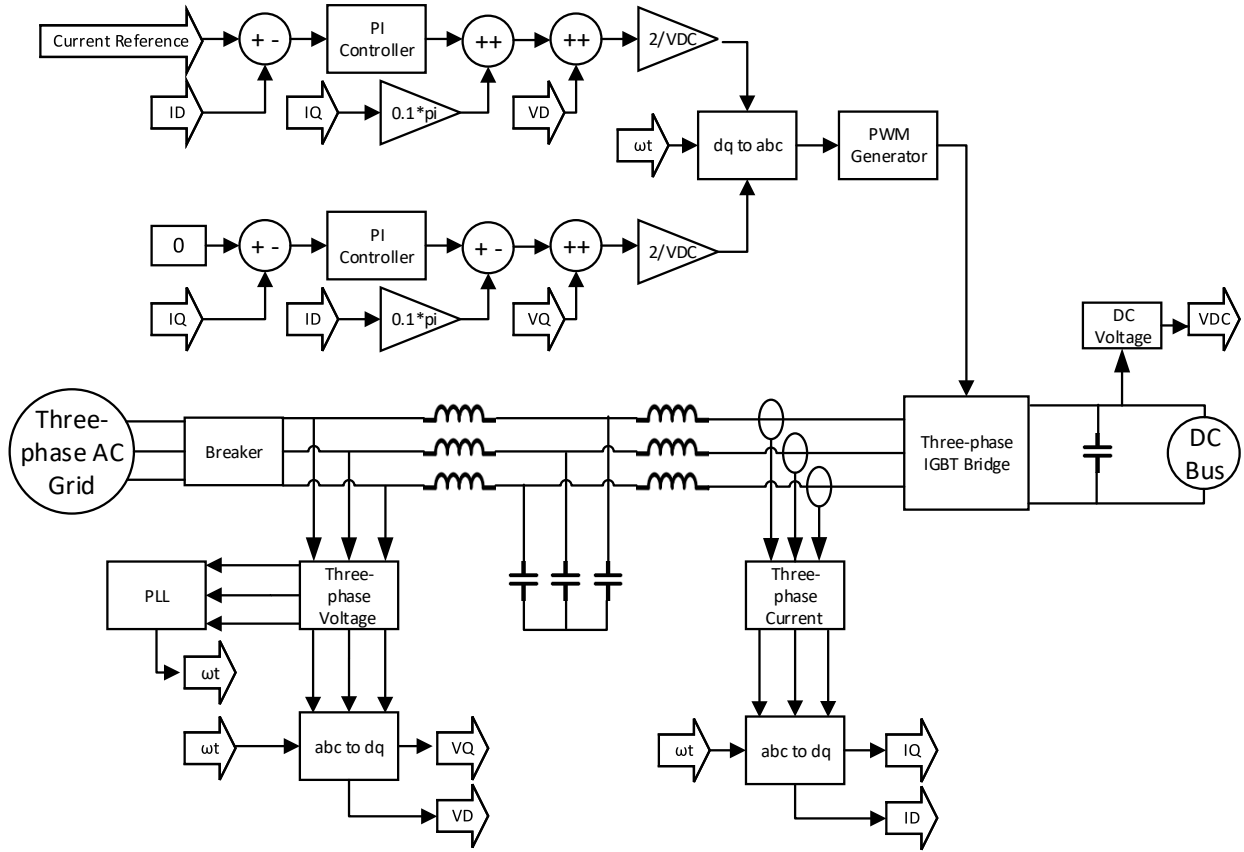


Fig. 6. The bidirectional grid connected converter system with its control architecture.

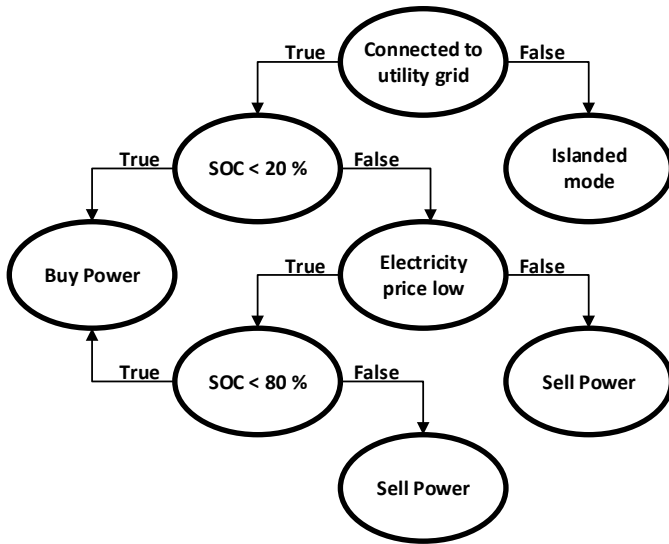


Fig. 7. Flowchart for the coordinated control.

A. Grid disconnecting and reconnecting

In the first simulation, the transitions to and from utility grid-connected operations are performed. At 0.3s the utility grid is disconnected from the microgrid to simulate the transition to an islanded microgrid that must function alone without external power support. At 0.6s the utility grid is connected back on and power is once again exchanged with

the utility grid. In Figure 8 the current exchanged between the utility grid and the microgrid is viewed. The amplitude and direction of the current and thus power is dictated by the coordinated control scheme depending on electricity price. At reconnection, a short transient period is seen as the phases of the current adjusts to the utility grid voltage phases to produce the set active power exchange.

The DC bus voltage is illustrated in Figure 9 and at the transition points in time the operation of the voltage controller is tested. Peak voltage fluctuations of about 1.4% are observed before the voltage is regulated back to its nominal value of 800 V. In Figure 10 the battery parameters are shown. The current exchange between the microgrid and the battery changes as the voltage controller regulates the DC bus voltage. In this simulation, the battery is providing power to the microgrid in the first and third periods as electricity is sold to the utility grid. During the middle period in islanded operation, the battery is instead charged as the power production by the PV system is larger than the consumption of 800 W in the microgrid load.

B. Grid connected operation

In the second simulation, the microgrid and utility grid is connected through the entire simulation period and instead two different electricity prices are tested that govern the power exchange. During the first half of the simulation, the electricity price is set to be low and the battery SOC to 50%. Thus, power

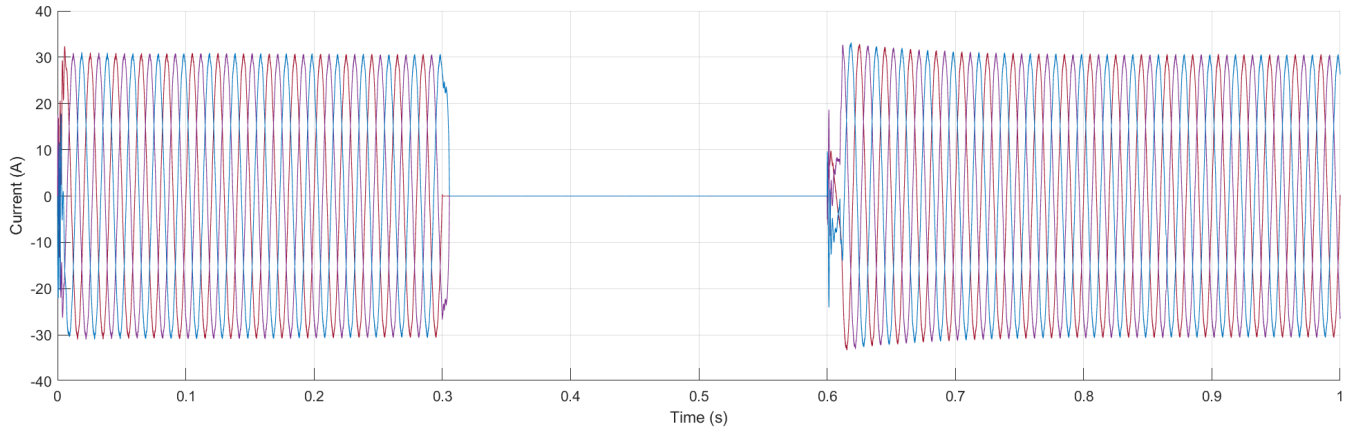


Fig. 8. The three-phase current exchange with the utility grid during the transitions between grid connected and isolated operation.

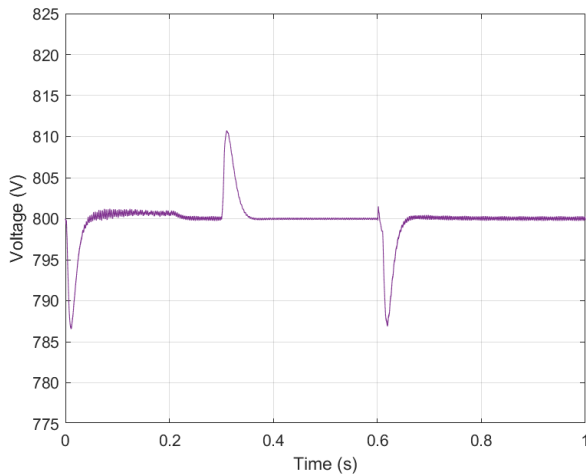


Fig. 9. The DC bus voltage showing the functionality of the voltage controller at disconnection and reconnection to the utility grid.

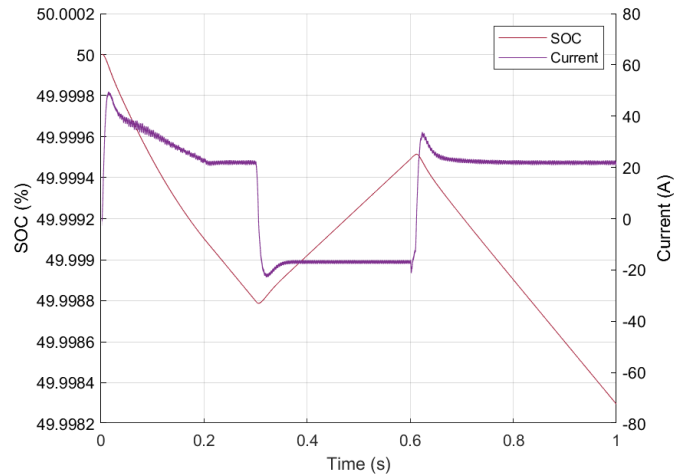


Fig. 10. Battery SOC and current exchange with the DC during at the transitions between grid connected and isolated operation.

flows into the microgrid from the utility grid and the battery is charged as shown in Figure 11. At 0.5 s the price changes to high, resulting in electricity being sold to the utility grid and discharging of the battery. The three-phase current is viewed in Figure 12 and at 0.5 s the change in direction is illustrated. The abrupt inversion of the current disturbs the DC bus voltage to a peak voltage fluctuation of 2.8% before the voltage controller regulates it back to its nominal value. This is seen in Figure 13.

C. Islanded operation

In the third simulation, the microgrid and utility grid is disconnected through the entire simulation period. Instead a varying irradiance on the PV system and a varying load connected to the microgrid test the capability of the isolated microgrid. The programmed irradiance reflects over to the power produced by the PV system seen in Figure 14. Together with a shifted peak power consumption illustrated in the same figure this simulation forces the battery to first be charged and later discharged to compensate for the difference in produced

and consumed power and maintain a stable DC bus of 800 V. The battery SOC and current are illustrated in Figure 15. The effect on the DC bus voltage is very slight throughout the entire simulation as seen in Figure 16.

V. DISCUSSION

The DC microgrid proposed in this paper consists of several parts designed to work both in connection with the grid and in isolated mode. The difficulty lies in making the system stable and also function in the transition between the two modes. Three main parts of the microgrid are interesting to look at to make it effective and stable namely the PV system, the grid converter system and the battery system.

For the PV system, the most important objective is to maximise power production to produce an efficient system. An unregulated system will not work at the voltage that is most effective when the irradiance and temperature are changing. To do this a MPPT algorithm is implemented. This ensures that the power production will be more efficient and therefore improve the quality of the microgrid. This can be seen in

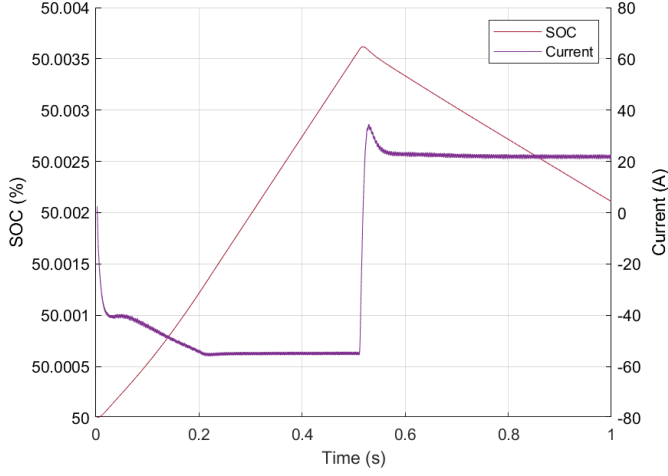


Fig. 11. The battery SOC and current when the electricity price goes from low to high.

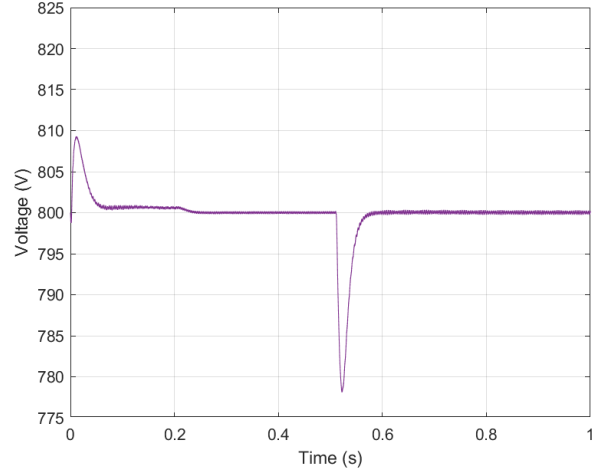


Fig. 13. The DC bus voltage when the electricity price goes from low to high.

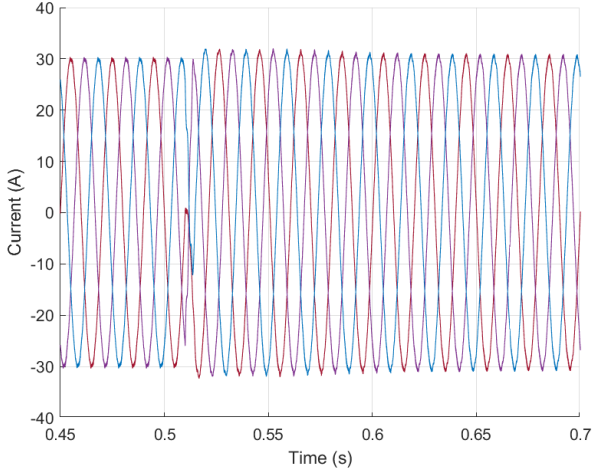


Fig. 12. The three-phase current exchange with the utility grid when the electricity price goes from low to high.

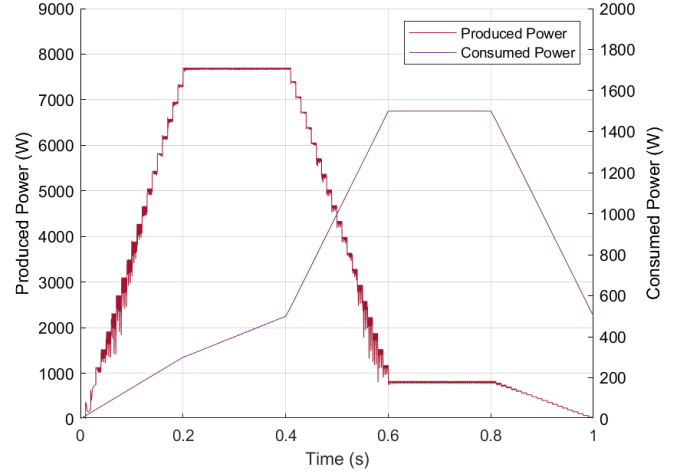


Fig. 14. Produced power by PV system and consumed power by DC load.

Figure 3 where the most effective voltage level for different irradiances is shown.

The grid converter system connects the DC bus of the microgrid with the utility grid. It is a three-phase bidirectional grid-connected inverter that makes it possible to transfer power both from the utility grid to the microgrid and the other way around. The converter is current controlled and it is therefore easy to alter the reference current when the power flow should change. When the microgrid is in isolated mode the grid converter system is not used because there is no connection with the utility grid. It is possible to make the grid converter voltage controlled so that this converter, instead of the battery converter, control the DC bus voltage. Such an implementation can be seen in [7].

The battery system consists of a controlled bidirectional buck-boost converter that controls the voltage of the DC bus. The buck-boost converter enables the battery to both charge and recharge depending on the setting. To stabilise the DC

bus voltage, the converter of the battery system is voltage controlled. The battery is therefore transferring power in the direction that is needed to keep the DC bus voltage stable. In isolated mode, this is a must since the utility grid is disconnected and can not control the voltage on the DC bus. In connected mode, the utility grid could keep the voltage of the DC bus stable if the grid converter system was voltage-controlled but with the battery always controlling the voltage, it is easier to keep the system stable in the transitions between the modes. The voltage of the DC bus at the transition between the modes can be seen in Figure 9.

VI. CONCLUSION

The full system is a complete working microgrid that can operate both in connection to the utility grid as well as in isolated mode. The PV system uses a MPPT algorithm to always operate on the voltage level that is most effective for the given irradiance and temperature. To enable power

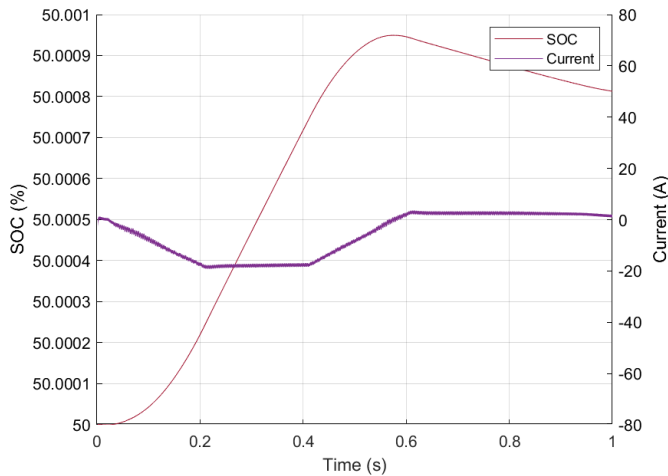


Fig. 15. The battery SOC and current when produced and consumed power are fluctuating.

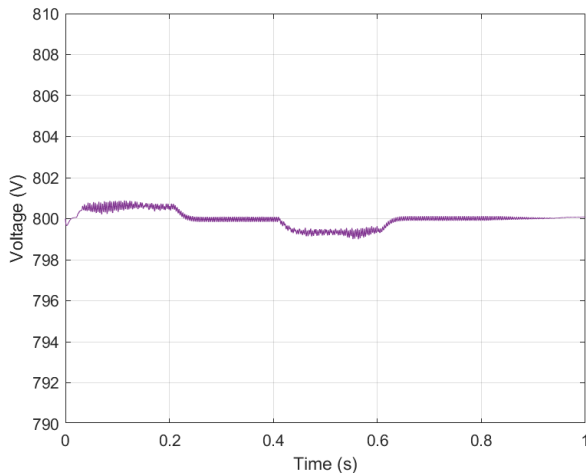


Fig. 16. The DC bus voltage when produced and consumed power are fluctuating.

trading with the utility grid, the grid inverter is a current-controlled three-phase bidirectional inverter. This also makes it straightforward to control the direction of the power. The battery system uses a bidirectional buck-boost converter that enables the battery to both be charged and discharged. The controller operates on a voltage reference so that the battery always controls the voltage on the DC bus.

For future work, it could be interesting to verify the simulations by implementing the system with hardware. It may also be interesting to add more components to the microgrid, such as a small wind turbine or an AC load.

APPENDIX A PRINTOUT OF DC MICROGRID SIMULINK SYSTEM

APPENDIX B PRINTOUT OF MATLAB CODE FOR SIMULINK MODEL

ACKNOWLEDGMENT

The authors would like to thank Qianwen Xu for her continued supervision, support and guidance throughout the project. The authors would also like to thank Mengfan Zhang for his assistance.

REFERENCES

- [1] United Nations. (2022, Apr.) The paris agreement. United Nations Framework Convention on Climate Change, Bonn, Germany. [Online]. Available: <https://unfccc.int/process-and-meetings/the-paris-agreement/the-paris-agreement>
- [2] Fiona Harvey. (2021, Nov.) What are the key points of the glasgow climate pact? The Guardian, London, United Kingdom. [Online]. Available: <https://www.theguardian.com/environment/2021/nov/14/what-are-the-key-points-of-the-glasgow-climate-pact-cop26>
- [3] D. J. Becker and B. Sonnenberg, "Dc microgrids in buildings and data centers," in *2011 IEEE 33rd International Telecommunications Energy Conference (INTELEC)*, 2011, pp. 1–7.
- [4] M. Nasir, H. A. Khan, N. A. Zaffar, J. C. Vasquez, and J. M. Guerrero, "Scalable solar dc micrigrids: On the path to revolutionizing the electrification architecture of developing communities," *IEEE Electrification Magazine*, vol. 6, no. 4, pp. 63–72, 2018.
- [5] E. Rodriguez-Diaz, J. C. Vasquez, and J. M. Guerrero, "Intelligent dc homes in future sustainable energy systems: When efficiency and intelligence work together," *IEEE Consumer Electronics Magazine*, vol. 5, no. 1, pp. 74–80, 2016.
- [6] 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 Transactions on Industrial Electronics*, vol. 64, no. 7, pp. 5930–5941, 2017.
- [7] R. Evode, "Modeling of electric grid behaviors having electric vehicle charging stations with g2v and v2g possibilities," in *2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, 2021, pp. 1–5.
- [8] S. Singh, S. Manna, M. I. Hasan Mansoori, and A. Akella, "Implementation of perturb amp; observe mppt technique using boost converter in pv system," in *2020 International Conference on Computational Intelligence for Smart Power System and Sustainable Energy (CISPSE)*, 2020, pp. 1–4.
- [9] F. Miyagishima, S. Augustine, O. Lavrova, H. Nademi, S. Ranade, and M. J. Reno, "Maximum power point tracking and voltage control in a solar-pv based dc microgrid using simulink," in *2021 North American Power Symposium (NAPS)*, 2021, pp. 1–4.
- [10] F. Blaabjerg, R. Teodorescu, M. Liserre, and A. Timbus, "Overview of control and grid synchronization for distributed power generation systems," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 5, pp. 1398–1409, 2006.
- [11] J. Rocabert, A. Luna, F. Blaabjerg, and P. Rodríguez, "Control of power converters in ac microgrids," *IEEE Transactions on Power Electronics*, vol. 27, no. 11, pp. 4734–4749, 2012.
- [12] G.-C. Hsieh and J. Hung, "Phase-locked loop techniques. a survey," *IEEE Transactions on Industrial Electronics*, vol. 43, no. 6, pp. 609–615, 1996.
- [13] A. Timbus, M. Liserre, R. Teodorescu, and F. Blaabjerg, "Synchronization methods for three phase distributed power generation systems - an overview and evaluation," in *2005 IEEE 36th Power Electronics Specialists Conference*, 2005, pp. 2474–2481.
- [14] C. J. O'Rourke, M. M. Qasim, M. R. Overlin, and J. L. Kirtley, "A geometric interpretation of reference frames and transformations: dq0, clarke, and park," *IEEE Transactions on Energy Conversion*, vol. 34, no. 4, pp. 2070–2083, 2019.

Context G: Power system planning and electricity markets

POPULAR DESCRIPTION

Plug in your electric car and make money

Anyone can play a part in solving the future power supply and demand problems by utilizing *Vehicle to Grid (V2G)* technology. By connecting electric cars to the electric grid and selling power stored in its battery when demand is high, V2G technology has the potential to smooth out power demand peaks. This, while at the same time generating cash to the car owner.

We are facing a bigger change than the industrial revolution: the third electrical revolution. All sectors will need to be electrified to fight climate change. Because of the unreliability associated with the dependency of the sun shining or the wind blowing, the future of renewable energy brings instability with it. A solution to this problem is found in the use of energy storage. Batteries are a prime candidate for the future and have the potential to store immense amounts of power, but producing them is resource intensive. This begs the question, why not utilize the full potential of available batteries?

In 2030 it is expected that electric cars will provide a total energy storage capacity of 7 TWh globally. 7 TWh is equivalent to the annual energy consumption of over 450 000 Swedish households. This shows that there is great potential for electric cars to balance the power grid, and that it can be used in the optimization of the electricity market. V2G technology could also remove pressure on the electrical grid when electricity demand is high and thereby lessen the need for grid expansion. V2G technology brings with it the opportunity to reduce the cost for the car owner as well. By making the purchasing of electricity more dynamic there will be advantages for the car owner.

The energy storage of the car battery lets the car owner buy electricity when demand and prices are low, and sell when the prices are high. This leads to less energy demand during high demand hours and more energy supply during low supply hours. This contributes to all consumers saving money on their next electricity bill. The result benefits both the car owner and the rest of the grid.

SUMMARY OF PROJECT RESULTS

Electric power production is a considerable source of CO₂ emissions, and therefore an important part of sustainable development is carbon neutral energy production. As intermittent renewables without any inertia like solar and wind power get an increasingly important role on the energy market, the electricity production will be less reliable. In order to combat this lack of reliability the development of a more reliable power grid, a more reliable energy market and more reliable regulating power are studied respectively in each project of this context.

In **project G1**, the goal was to investigate the impact the introduction and placement of wind power has on the voltage stability of a power system in Jämtland, Sweden. The voltage stability investigation was done using load flow analysis, which is an application of numerical methods for solving nonlinear systems of equations. Additionally, the introduced wind power's effects on the reactive power levels in the system were studied. Furthermore, the possible use of solar power as a reactive power producer in cities was analyzed as a substitute for reactive compensators, which often are necessary in order to maintain acceptable voltage levels. The results of the project give insight into how the voltage levels in the power system behave and what kind of reactive power control systems are necessary when renewable and intermittent energy sources are introduced to the grid. However, this study does not take into consideration the more realistic environmental, social and cultural elements, which could affect the placement of wind power. In reality there are more complex situations when

introducing a wind power plant and more things to consider when making decisions. Not only are there environmental attributes to consider, but also having a dialogue with the people living in the area of construction is important. Studies examining the negative and positive effects on the surrounding world would enhance the results.

In project G2, the objective was to examine the different ways to incentivize investments in capacity when the penetration of renewable energy sources in the system becomes higher. Capacity is a term used for electricity that can be generated in a short notice; one could think of it as a battery for the system. In a capacity market, the producers do not only get paid by energy produced, they get paid for being on stand-by. The inadequacy of capacity increases with larger shares of renewables on the energy market. Solar and wind power have lower reliability compared to more resilient power producers such as nuclear power. With this background, one can argue that the market should not evolve around energy, but instead around the producer's capacity.

Group G2a conducted a review of a method in which a separate market for capacity is used. In this market, power plants participate in an auction in which the participants get paid to be available in a three year contract. The review work was focused on the energy market in the U.S, and especially the network operator PJM. Group G2b focused on the European market instead, specifically Sweden and how effective the use of strategic reserves ensure reliable and sufficient capacity. The strategic reserve approach pays side-lined power plants to ensure capacity during periods when energy supply is extraordinarily scarce.

Today, consumers of electricity have limited ability to know exactly when the demand and electricity prices are high and when the demand and prices are low. It is also hard for consumers to manually react to changes in electricity prices, like waking up in the middle of the night when the prices are low to do laundry. Advancements in IT and electronics related technology over the past years has enabled consumer's ability to react to price-variability and is expected to develop and improve further over the coming decades. A proposal for further research is to investigate the need for a capacity market in a power system with improved responsivity from the demand side. Furthermore, an investigation on energy storage to take advantage of favorable weather conditions is suggested as an additional research topic. If we were able to store energy when the production is high, we could counteract the unreliability that renewable sources bring to the system, which could in the end lead to an energy system with 100% renewable sources.

In project G3, the focus was on hydropower plants and their ability to regulate electricity production. Hydropower can be planned in a way that the generation of electricity answers to the demand rather than how much water flows in the river continuously. However, there is a limit to which extent this can be done and it is of interest to work out how much we can rely on hydropower to compensate for other renewable energy sources. In this project, the Skellefteå river was modeled with the newly developed software Spine. The aim was to optimize future electricity production, and investigate the regulating capacity in the river's hydropower plants. To make the model applicable, the aim was to implement a piecewise linear function of the electricity production in the model.

For future projects, it would be interesting to implement several rivers in the model and investigate the regulating capacity with the piecewise linear model, since in Spine it has so far mostly been done with linear dependencies. With the knowledge of how well all the Swedish rivers can cover a changing demand of energy, it would be easier to tell how much we can rely on hydropower on a bigger scale.

IMPACT ON SOCIETY AND ENVIRONMENT

Increasing global electricity demand is inevitable and incentives for carbon neutral electricity production is important to mitigate the effects of global warming. Furthermore, electricity should be accessible to everyone and work within this context can contribute to this goal. The research within power system planning enables the development of renewable energy on the electric grid. Moreover, it also contributes to the grid functioning in a manner that makes electricity accessible. These effects are especially important as the electrical revolution, in both the industry and the transportation sector, is a fact.

Wind power offers cheap and CO₂-free energy, and in this way it contributes globally in a positive way by combating climate change. There are, however, negative effects on the local environment associated with wind power. The wind parks take up

space in nature and the increase of human activity during construction and maintenance may disturb wildlife. In order to ensure that animal populations are not harmed, the impact a wind park will have on the local environment must be thoroughly assessed in the planning phase. If this is not properly done critical animal populations may be harmed beyond saving. Forcing constructors to assess the effects wind power parks have on wildlife before constructing them will narrow down location options and add costs in the planning phase, and may result in lessening the attractiveness of the wind power construction market. At the same time demanding a stricter wildlife assessment may spur innovation in wind turbine and wind park design with more wildlife safe wind power as an outcome.

Compared to fossil energy sources, hydropower has a relatively small negative impact on the environment. It mainly affects the surrounding land areas and ecosystems locally. For example, the surrounding ecosystems may be affected by changing water flow, and migration routes for fish and other migrating species can be blocked. It creates problems for the fish when it comes to procreation and finding food, as well as having a negative impact on the living conditions of plants. Overall, the local water system's ability to offer different ecosystem services decreases which means less utilities for individuals and society. Fish farms and fish ladders are solutions that decrease the impact of these problems. Even if there are negative consequences on ecosystems, the fact that hydropower is a considerable part of Sweden's electricity system indicates that the advantages we get with it are greater.

The higher the amount of renewables, the lower the stability of the grid becomes. However, we still want investors to invest in green generation. We are dependent on good weather conditions to provide adequate generation, which is not always possible. Therefore, there still exists a need for older ways of generating electric power such as coal and nuclear power plants, to protect the integrity of the system. This is where the capacity markets take place; to ensure that enough capacity is in place to pave the way for investments in green technology. However, some could argue that it is paradoxical to enable generation of electricity using fossil fuel to stay in the system when the goal is to transition to renewable energy production. Nevertheless, a capacity market would make it possible for more renewables to be built than without it. To be able to completely abandon the old generation of power plants would require technological innovations to take place. For example, we would have to be able to store left-over power from renewable energy sources during optimal weather conditions.

Since all projects in this context enable the development of renewable energy, especially wind, the ethical aspects of wind power should be discussed. The placement of wind power parks is well debated. Both the view of the wind parks and the sound pollution that comes from wind parks underlie the discussion of where these should be placed. The question is whether one can justify deforestation for a wind park and if it is defensible to place a wind park in areas important for the natives.

A question that arises is; how much can we charge customers for electricity? The design of capacity markets is closely related to a price-cap on the electrical prices. Without a price-cap, the market would behave as a perfect market; when prices rise too high, demand will drop and people would voluntarily refrain from consuming. We can agree that electricity has become a necessity for society and therefore the cost of electricity for consumers should be reasonable. For residential consumption of electricity, the major consumption comes from necessities like washing and cooking. Therefore, in many countries, the government has deemed it unethical to let the market decide the prices, which has led to governments enforcing a price-cap. However, a high price-cap could lead to customers being unable to afford electricity at times of high prices. A low price-cap would instead lead to power plants being decommissioned because of unprofitability. Even if we implement a market for capacity, we should aspire to do so in a price wise, ethical way.

Electricity demand can lead to national relationships and dependence on trade with questionable regimes. This can for example be seen in the EU's dependence on imported Russian energy. Fortunately, the shift to renewable electricity generation and these types of relationships go somewhat hand in hand. On the other hand investment in wind- and solar energy requires certain conflict minerals needed in the construction of the wind turbines and solar panels. Hence, while renewable energy trends offer opportunities in diminishing trade-dependence with oil, gas and uranium exporters, it might increase trade with certain mineral and metal exporters, which are not uncommonly hosted by non-ethical authorities. A desirable outcome is that through power system planning and market design achieve a power system with the least amount of dependence on problematic trade. Some fossil fuels and minerals are well known to be the root of trade conflicts with dictatorships and questionable democracies that do not value human rights.

Voltage Stability and Reactive Power - Introduction of Intermittent Renewable Energy Sources in a Power System

Erik Hagström and Tobias Jansson

Abstract—The electricity demand increases rapidly, and in order to mitigate climate change the power production needs to be renewable and free from green house gas emissions. When solar and wind power are introduced in the system, voltage instability might become a problem. This study aims to investigate voltage stability and the effects of reactive power compensation. It is done by performing power flow analysis on a simulated power system model in Jämtland, Sweden, with a large share of wind power and a relatively small share of sun power. The simulations are made in MATPOWER (MATLAB). The results reveal that the voltage levels in this study remain stable, with the reactive power being the limiting factor. The use of passive reactive power compensators, like shunt reactors, does not keep reactive power levels in the system within set limits. This study shows that in order to achieve that, active reactive power compensators are required.

Sammanfattning—Efterfrågan av elektricitet ökar snabbt, och för att kunna mildra klimatförändringarna behöver kraftproduktionen vara förnybar och fri från växtusgasutsläpp. När sol- och vindkraft introduceras kan spänningsstabilitet bli ett problem. Denna studie ämnar att undersöka spänningsstabilitet och effekterna av reaktiv effekt-kompensering. Det görs genom att utföra belastningsfördelningsberäkningar på en simulerad kraftsystemmodell i Jämtland i Sverige, med en stor andel vindkraft och en relativt liten andel solkraft. Simuleringarna görs i MATPOWER (MATLAB). Resultaten visar att spänningsnivåerna i denna studie hålls stabila, där reaktiv effekt är den begränsande faktorn. Användning av passiva reaktiv effekt-kompensatorer, såsom shuntreaktorer, håller inte de reaktiva effektnivåerna inom önskade gränser. Denna studie visar att, för att kunna uppnå det, så krävs det aktiv reaktiv effekt-kompensering.

Index Terms—Voltage Stability, Reactive Power, Renewable Energy, Wind Power, Power Flow Analysis, MATPOWER.

Supervisors: Evelin Blom and Lennart Söder

TRITA number: TRITA-EECS-EX-2022:144

I. INTRODUCTION

A. Background

Global electricity demand is expected to double by 2060 [1]. In order to meet this demand and at the same time fulfill the Paris Agreement to limit the global average temperature rise of 1.5 °C, the energy production needs to be clean and renewable [2]. A large share of renewable energy sources like solar and wind power will therefore need to be introduced in the power system. They are energy sources that will help mitigate climate change and contribute to emission-free generation of electricity. However, their dependency of the weather makes them intermittent and there will simply be no electricity

produced when the wind does not blow or the sun does not shine [3]. Furthermore, the electricity produced needs to be consumed at an instant, with no capabilities of buffer capacity or storage. As today's modern society is highly dependent on electricity around the clock, it can be difficult to match the demand and weather dependent production. When these two do not match, there will be frequency deviations. This can lead to voltage deviations as the power flow across the transmission system changes [4]. One of the challenges is therefore to keep the voltage levels within acceptable limits.

Controlling the voltage levels to avoid disruption or damage to the equipment is consequently crucially important. When voltage deviates more than a set value, the equipment connected could be damaged. A voltage much higher than the nominal voltage of the equipment can lead to higher losses, lower efficiency and potentially a reduction of equipment lifetime. If the voltage is much lower than the nominal voltage it can cause interruptions or reduce the strength of the equipment, e.g. a motor's ability to produce torque will decrease [5].

Transmitted power consists of active and reactive power. Reactive power plays an important role when it comes to voltage stability and is necessary in order to distribute active power [6]. Under low load conditions transmission lines produce reactive power [7] and this may result in an undesired amount of reactive power in the power system. In order to regulate the reactive power levels of the system, shunt capacitors and shunt reactors are used [6]. They can be installed in the nodes of the system and produce/consume reactive power respectively. Shunt capacitors and reactors are passive elements of the power system, and the only way of controlling them is by turning them on or off [6]. This results in a step-wise regulation with the consequence of reduced accuracy. Another aspect affecting the accuracy of the reactive power compensation provided by shunt capacitors and reactors is the time response. A faster response enables a more even regulation, something that automatic and more flexible reactive power compensators can provide [8].

Having reactive power production near loads that consume reactive power is advantageous due to line losses that occur when reactive power is transferred over long distances [9]. Most of the solar power installed are placed on rooftops, which places them close to the loads. The system that converts light into electricity, solar photovoltaic (PV) power systems, can both produce and consume reactive power [10]. This makes it a candidate for reactive power compensation. This might

at the same time reduce the need to add shunt capacitors to increase the reactive power [11].

When establishing large shares of wind power into the system, several parameters need to be considered. The placement of the wind power plants not only have to take wind speeds in consideration, but also the local environment, in order to have an ecologically sustainable development. The network structure also affects the distribution of power and plays a key role in achieving voltage stability [12]. This project aims to investigate if the voltage levels can be maintained within $\pm 10\%$ of a set value when wind power is introduced, and if solar power can provide reactive power compensation.

B. Goals

In this project the *main goal* is to set up a fictive power system with hydropower plants, wind power parks and load centers in the form of cities, working within $\pm 10\%$ voltage magnitude deviation from the set value. The goal is to introduce wind power while still keeping the system within the voltage limits. This is done by creating a model of a power grid in Jämtland. *Another goal* is to keep the magnitude of the reactive power production/consumption in the hydropower plants smaller than 5% of the active power levels, and investigate the potential solar power has to provide reactive power compensation. The aim is to show how reactive power compensation can affect the voltage levels in the power system. A *further goal* is to show how the transmission line grid structure affects the voltage limits when introducing renewable power production.

This paper does not aim to simulate the real grid in Jämtland, but instead aims to model a future power system, in order to analyze what is important when designing a grid with high amounts of wind power.

II. THEORY

In this section power flow analysis is described, as this is the method used for analyzing the voltages, power flows and losses of the simulated power system. To begin with, some theory explaining the characteristics of a symmetric three phase power system is presented, as these are needed for the power flow calculations. Additionally, the method of how to perform the power flow analysis is described.

A. Power Lines

The electric power system can at a large scale be viewed as nodes of production and consumption centers representing power plants and cities respectively, connected via transmission lines. Three phases are used in the transmission lines, leading to that three conductors are used. This allows for a more even and higher power transmission than if only one phase is used. The load is assumed to be symmetric in this article, meaning the load is evenly distributed on each phase [9].

Transposing the three phases of the transmission line as shown in Fig. 1 results in an equal average distance to the ground and the other conductors, for every conductor.

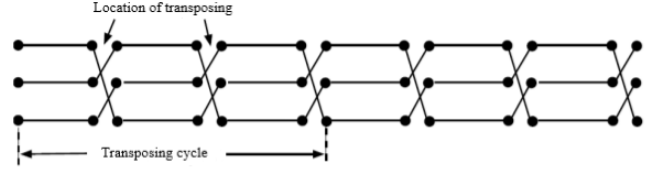


Fig. 1. Representation of a transposed three phase transmission line [9].

The transposition of the three phases yields a more evenly distributed inductance, which is one of the physical quantities of a transmission line impacting the power flow, voltage levels and losses in the power system. All equations described in this section are valid for a transposed three phase transmission line under symmetrical conditions. These physical quantities are described per-unit of length, often per kilometer, and a symmetric three phase power line can be represented by Fig. 2 where r denotes the resistance of the line, l the inductance, c the capacitance and g the conductance [9].

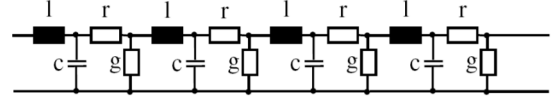


Fig. 2. Representation of a symmetric three phase transmission line [9]

The line's resistance is described by equation (1)

$$r = \frac{\rho}{A} [\Omega/km] \quad (1)$$

where ρ denotes the resistivity of the conducting material, and A is the conductors cross-sectional area. The inductance is expressed in equation (2).

$$l = 2 \cdot 10^{-4} \cdot \left(\ln \frac{a}{d/2} + \frac{1}{4n} \right) [H/km, phase] \quad (2)$$

n is the number of conductors per phase, a is the geometric average distance as described by equation (3) and Fig. 3, and d represents the diameter of the conductor.

$$a = \sqrt[3]{a_1 a_2 a_3} [m] \quad (3)$$

Equation (4) describes the capacitance of the three phase line.

$$c = \frac{10^{-6}}{18 \cdot \ln \left(\frac{2H}{A} \cdot \frac{a}{d/2} \right)} [F/km, phase] \quad (4)$$

H and A are calculated in equation (5) and (6) respectively, with the geometric quantities shown in Fig. 3. H is the geometric mean height of the conductors, and A is the geometric mean distance between the conductors and their mirror image.

$$H = \sqrt[3]{H_1 H_2 H_3} [m] \quad (5)$$

$$A = \sqrt[3]{A_1 A_2 A_3} [m] \quad (6)$$

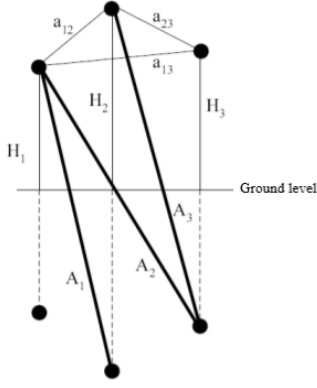


Fig. 3. Cross section of a transmission line and its mirror image [9].

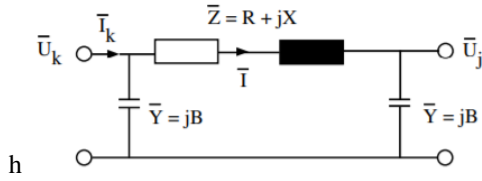


Fig. 4. Transmission line Π-model [11].

The conductance is in most applications omitted as it is heavily dependent on conditions such as humidity and air pollution, and there is no reliable data describing the phenomena. Accordingly, the conductance will not be taken into consideration in this article.

When performing calculations on the power system, every transmission line is modeled as shown in Fig. 4. Placing the impedance \$\bar{Z}\$ along the power line, and half the admittance \$\bar{Y}\$ between the power line and earth at each end of the power line results in a Π-model.

The impedance \$\bar{Z}\$ and admittance \$\bar{Y}\$ depend on the lines physical quantities according to equations (8) and (9) where \$s\$ is the length of the transmission line in kilometers and \$j\$ is the imaginary unit described in equation (7).

$$j = \sqrt{-1} \quad (7)$$

$$\bar{Z} = R + jX = (r + jx) \cdot s \text{ } [\Omega/\text{phase}] \quad (8)$$

$$\bar{Y} = jB = j \frac{bs}{2} \text{ } [\Omega/\text{phase}] \quad (9)$$

With the frequency \$f\$, the reactance and susceptance \$x\$ and \$b\$ are calculated using the inductance \$l\$ and capacitance \$c\$ in equations (10) and (11).

$$x = 2\pi \cdot f \cdot l \text{ } [\Omega/\text{km}, \text{phase}] \quad (10)$$

$$b = 2\pi \cdot f \cdot c \text{ } [\Omega/\text{km}, \text{phase}] \quad (11)$$

Now the voltage at each end of the line shown in Fig. 4 can be calculated using equations (12), (13) and (14) [11].

$$\bar{U}_j = \bar{U}_k - \sqrt{3\bar{Z}\bar{I}} \quad (12)$$

$$\bar{I} = \bar{I}_k - \bar{Y} \frac{\bar{U}_k}{\sqrt{3}} \quad (13)$$

$$\bar{U}_j = (1 + \bar{Z}\bar{Y})\bar{U}_k - \sqrt{3\bar{Z}\bar{I}_k} \quad (14)$$

B. Active Power, Reactive Power and Power Factor

The power transmitted in a power system consists of a real and an imaginary part. These are called the active power and the reactive power and together they form the complex power in equation (15) [13].

$$\bar{S} = P + jQ \text{ } [VA] \quad (15)$$

The apparent power is the absolute value of the complex power shown in equation (16) and Fig. 5.

$$|\bar{S}| = \sqrt{P^2 + Q^2} \text{ } [VA] \quad (16)$$

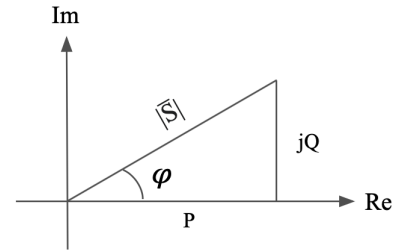


Fig. 5. Power triangle [9].

The power factor is the relationship between the real power (\$P\$) and apparent power \$|\bar{S}|\$, with the angle \$\varphi\$ between them, as shown in equation (17).

$$\cos \varphi = \frac{P}{|\bar{S}|} \quad (17)$$

If \$\cos \varphi < 1 \Rightarrow\$ voltage and current are not in phase and higher currents needs to be transmitted to obtain the same power, which can lead to higher transmission losses. Thus, it is desirable to have a high power factor [9].

C. Per-Unit System

The per-unit system is commonly used in power systems to express voltages, currents, powers and impedances. One of the advantages of using the per-unit system is that the voltage drop, in percent, can be obtained immediately. It is a relation between the real value and a reference, or base value, as shown in equation (18). Another advantage is that calculations can be done with multiple voltage levels in the system [11].

$$\text{Per-unit value} = \frac{\text{Real value}}{\text{Base value}} \quad (18)$$

From a base voltage

$$U_{base} = \text{Main voltage} = \text{Base voltage [kV]} \quad (19)$$

and a complex base power

$$\bar{S}_{base} = \text{Three phase base complex power [MVA]} \quad (20)$$

the base current, shown in equation (21)

$$\bar{I}_{base} = \frac{\bar{S}_{base}}{\sqrt{3}U_{base}} [kA] \quad (21)$$

and base impedance, shown in equation (22)

$$\bar{Z}_{base} = \frac{U_{base}^2}{\bar{S}_{base}} [k\Omega] \quad (22)$$

can be calculated.

D. Power Flow Analysis

With power flow analysis the voltage magnitudes and voltage angles in a power system can be found. If the magnitude of the voltages and the phase angles in all the buses are known, the losses and information on how the transmission lines are loaded can be found. The voltage information, along with current flows and power flows, is derived from the power production and consumption in the system by formulating a system of equations. The system of equations can be solved with Newton-Raphson's method [9]. In power system analysis every node is referred to as a *bus*, and every bus is associated with four variables. These variables are the voltage magnitude U , the voltage phase angle θ , the active power P injected in the bus and the reactive power Q injected in the bus. In power flow analysis three different types of buses are used, with every type having two of the four variables defined [14]. At the start of a power flow analysis every bus needs to be classified as one of the three following types:

- **PQ-bus:** Net generation of active power and reactive power are known, voltage and phase angle are unknown. It represents a point in the power system where the power consumption can be considered as independent of the voltage.
- **PU-bus:** Net generation of active power and the voltage are known, net generation of reactive power and phase angle are unknown.
- **Slack-bus (U θ -bus):** Reference angle and voltage magnitude known, net generation of active and reactive power are unknown. Only one slack-bus exists within a system. This bus balances the power distribution.

As a means to perform the desired power flow analysis and calculate the voltages at every bus, the net active and reactive power productions of the system are required. This in turn requires the injected current in every bus and a useful tool

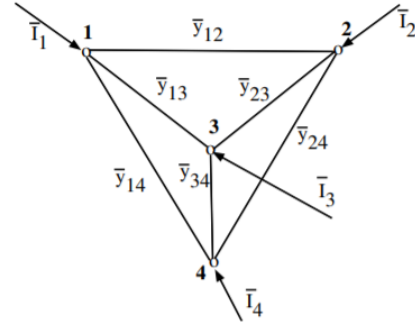


Fig. 6. Four bus power system [11].

for acquiring these is the admittance matrix \mathbf{Y} in combination with (13) and (14). Fig. 6 illustrates an example of a power system with four buses.

Assuming that the currents \bar{I}_1 , \bar{I}_2 , \bar{I}_3 and \bar{I}_4 are injected from an external source, and denoting the voltages at each bus by \bar{U}_1 , \bar{U}_2 , \bar{U}_3 and \bar{U}_4 , balance equations (23) and (24) are set up for bus 1. The admittance for each transmission line is represented in the figure as \bar{y}_{jk} where jk are the indices of the connected buses.

$$\bar{I}_1 = \bar{y}_{12}(\bar{U}_1 - \bar{U}_2) + \bar{y}_{13}(\bar{U}_1 - \bar{U}_3) + \bar{y}_{14}(\bar{U}_1 - \bar{U}_4) \quad (23)$$

$$\begin{aligned} \bar{I}_1 &= (\bar{y}_{12} + \bar{y}_{13} + \bar{y}_{14})\bar{U}_1 - \bar{y}_{12}\bar{U}_2 - \bar{y}_{13}\bar{U}_3 - \bar{y}_{14}\bar{U}_4 \\ &= \bar{Y}_{11}\bar{U}_1 + \bar{Y}_{12}\bar{U}_2 + \bar{Y}_{13}\bar{U}_3 + \bar{Y}_{14}\bar{U}_4 \end{aligned} \quad (24)$$

Expressing these equations for every bus enables the construction of the admittance matrix, shown in equation (25). The general admittance matrix for a system with n buses is described by equation (26) [11].

$$\mathbf{I} = \begin{bmatrix} \bar{I}_1 \\ \bar{I}_2 \\ \bar{I}_3 \\ \bar{I}_4 \end{bmatrix} = \begin{bmatrix} \bar{Y}_{11} & \bar{Y}_{12} & \bar{Y}_{13} & \bar{Y}_{14} \\ \bar{Y}_{21} & \bar{Y}_{22} & \bar{Y}_{23} & \bar{Y}_{24} \\ \bar{Y}_{31} & \bar{Y}_{32} & \bar{Y}_{33} & \bar{Y}_{34} \\ \bar{Y}_{41} & \bar{Y}_{42} & \bar{Y}_{43} & \bar{Y}_{44} \end{bmatrix} \begin{bmatrix} \bar{U}_1 \\ \bar{U}_2 \\ \bar{U}_3 \\ \bar{U}_4 \end{bmatrix} = \mathbf{Y}\mathbf{U} \quad (25)$$

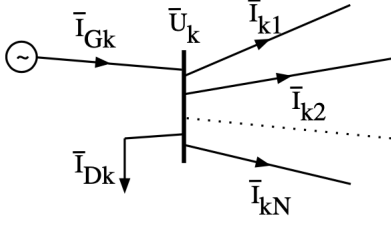
$$\mathbf{I} = \begin{bmatrix} \bar{I}_1 \\ \vdots \\ \bar{I}_n \end{bmatrix} = \begin{bmatrix} \bar{Y}_{11} & \dots & \bar{Y}_{1n} \\ \vdots & \ddots & \vdots \\ \bar{Y}_{n1} & \dots & \bar{Y}_{nn} \end{bmatrix} \begin{bmatrix} \bar{U}_1 \\ \vdots \\ \bar{U}_n \end{bmatrix} = \mathbf{Y}\mathbf{U} \quad (26)$$

Fig. 7 shows a representation of a bus, k , assuming symmetric three-phase. \bar{I}_{Gk} is the current supplied from the generator, \bar{I}_{Dk} is the current drawn from the load in the bus, $\bar{I}_{k1}, \bar{I}_{k2}, \dots, \bar{I}_{kN}$ are the currents that flow to the other buses.

According to Kirchhoff's first law, which states that the sum of the currents flowing into a junction is equal to the currents flowing out of a junction, it can be described as in equation (27)

$$\bar{I}_{Gk} - \bar{I}_{Dk} = \sum_{j=1}^N \bar{I}_{kj} \quad (27)$$

In other words, the sum of all the currents to bus k must be zero. \bar{I}_{kj} represents the current from bus k to bus j .

Fig. 7. Bus k in a system [9].

By conjugating equation (27) and multiplying it with the bus voltage, the injected currents are used to calculate the net active and reactive power productions in the following, shown in equation (28).

$$\bar{U}_k \bar{I}_{Gk}^* - \bar{U}_k \bar{I}_{Dk}^* = \sum_{j=1}^N \bar{U}_k \bar{I}_{kj}^* \quad (28)$$

Rewriting equation (28) as an expression for complex power according to equation (29)

$$\bar{S}_{Gk} - \bar{S}_{Dk} = \sum_{j=1}^N \bar{S}_{kj} \quad (29)$$

where the complex power produced by the generator in equation (30)

$$\bar{S}_{Gk} = P_{Gk} + jQ_{Gk} \quad (30)$$

the complex power consumed by the load in equation (31)

$$\bar{S}_{Dk} = P_{Dk} + jQ_{Dk} \quad (31)$$

and the transmitted power to bus j , in equation (32), are specified.

$$\bar{S}_{kj} = P_{kj} + jQ_{kj} \quad (32)$$

This gives the net production of active power in bus k , shown in equation (33)

$$P_{GDk} = P_{Gk} - P_{Dk} = \sum_{j=1}^N P_{kj} \quad (33)$$

and net production of reactive power in bus k , described in (34)

$$Q_{GDk} = Q_{Gk} - Q_{Dk} = \sum_{j=1}^N Q_{kj} \quad (34)$$

Once all of the buses have been classified, the admittance matrix Y has been produced and the net production of active and reactive power P_{GDk} and Q_{GDk} have been calculated, an approximation of the still unknown variables is done. These unknown variables are the phase angles θ for every PQ-bus and PU-bus, and the voltage magnitude U for every PQ-bus. This approximation can be set to $U = 1$ and $\theta = 0$ [13].

The injected power in every bus is then calculated in equations (35) and (36).

$$P_k = \sum_{j=1}^n P_{kj} \quad (35)$$

$$Q_k = \sum_{j=1}^n Q_{kj} \quad (36)$$

P_{kj} and Q_{kj} can be derived from equation (32) where

$$\begin{aligned} \bar{S}_{kj} &= \bar{U}_k (\bar{I}_{kj0}^* + \bar{I}_{kj0}) = \bar{U}_k \left(\bar{U}_k^* Y_{kj}^* + \frac{\bar{U}_k^* - \bar{U}_j^*}{\bar{Z}_{kj}} \right) = \\ &= U_k^2 (-jB) + \frac{U_k^2}{R - jX} - \frac{U_k U_j}{R - jX} e^{j(\theta_k - \theta_j)} = \\ &= U_k^2 (-jB) + \frac{U_k^2}{Z^2} (R + jX) \\ &\quad - \frac{U_k U_j}{Z^2} (R + jX) (\cos \theta_{kj} + j \sin \theta_{kj}) \end{aligned} \quad (37)$$

By dividing equation (37) into its real and imaginary part, the following expressions are obtained:

$$P_{kj} = \frac{U_k^2}{Z^2} R + \frac{U_k U_j}{Z^2} (X \sin \theta_{kj} - R \cos \theta_{kj}) \quad (38)$$

$$Q_{kj} = -BU_k^2 + \frac{U_k^2}{Z^2} X - \frac{U_k U_j}{Z^2} (R \sin \theta_{kj} + X \cos \theta_{kj}) \quad (39)$$

E. Solving power flows with Newton-Raphson's method

For every bus in the system there must be power balance. Equation (33) and (34) shows that the net production of active and reactive power must be equal to the active and reactive power transmitted from bus k to the other buses. As the calculations that have been done so far are made with approximations of some of the variables, the system might not yet be in the state of power balance. In order to check if the system is in balance, the difference between the injected power and the power production of each bus is computed as shown in equation (40) [13].

$$\begin{cases} \Delta P = P_{GDk} - P_k \\ \Delta Q = Q_{GDk} - Q_k \end{cases} \quad (40)$$

If the difference is greater than the accepted error, a better approximation needs to be made. This is done with Newton-Raphson's method.

The first step when solving a power flow problem with this method is creating the admittance matrix in order to calculate the net productions P_{GDk} and Q_{GDk} in equations (33) and (34).

In the second step the injected power in each bus is calculated according to equations (35) and (36). This is where the solution to the power balance is checked as described in equation (40).

The third step determines how the voltage magnitudes and angles are to be changed in order to get closer to the correct

solution. Now the Jacobian of the system is calculated. The Jacobian contains the partial derivatives of the active and reactive power functions, with respect to the voltage magnitude and voltage angle, in the buses. The structure of the jacobian is shown in equation (41).

$$JAC = \begin{bmatrix} H & N \\ J & L \end{bmatrix} \quad (41)$$

T is the total number of buses in the system, M is the number of PU-buses and there is one slack bus.

- H is a $(T-1) \times (T-1)$ matrix
- N is a $(T-1) \times (T-M-1)$ matrix
- J is a $(T-M-1) \times (T-1)$ matrix
- L is a $(T-M-1) \times (T-M-1)$ matrix

where

- $H_{kj} = \frac{\delta P_k}{\delta \theta_j}$ $k \neq \text{slack bus}$ $j \neq \text{slack bus}$
- $N_{kj} = \frac{\delta P_k}{\delta U_j}$ $k \neq \text{slack bus}$ $j \neq \text{slack bus}$ and PU-bus
- $J_{kj} = \frac{\delta Q_k}{\delta \theta_j}$ $k \neq \text{slack bus}$ and PU-bus $j \neq \text{slack bus}$
- $L_{kj} = \frac{\delta Q_k}{\delta U_j}$ $k \neq \text{slack bus}$ and PU-bus $j \neq \text{slack bus}$ and PU-bus

Step four updates the voltage magnitudes and angles. With the Jacobian matrix containing the applicable partial derivatives of the buses, it is possible to make a better approximation of the variables. This is done by matrix multiplication of the inverse of the Jacobian matrix, and a vector containing the errors ΔP and ΔQ as shown in equation (42). The approximations are then updated in equation (43).

$$\begin{bmatrix} \frac{\Delta \theta}{U} \end{bmatrix} = \begin{bmatrix} H & UN \\ J & UL \end{bmatrix}^{-1} \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} \quad (42)$$

$$\begin{aligned} \theta_k &= \theta_k + \Delta \theta_k & k \neq \text{slack bus} \\ U_k &= U_k \left(1 + \frac{\Delta U_k}{U_k} \right) & k \neq \text{slack bus and PU-bus} \end{aligned} \quad (43)$$

Now the calculations return to step two. This process of calculating the injected power, determining the differences ΔP and ΔQ and updating the variables is now performed until the error checked in step two is small enough. Once this is done, the generated powers in the slack bus can be determined from equation (44)

$$\begin{cases} P_k - P_{GDk} = 0 \\ Q_k - Q_{GDk} = 0 \end{cases} \quad (44)$$

The power flows are described by equations (38) and (39). The power line losses can be calculated using equations (45) and (46) [13].

$$P_{Lkj} = P_{kj} + P_{jk} \quad (45)$$

$$Q_{Lkj} = Q_{kj} + Q_{jk} \quad (46)$$

F. Transmission Line Losses

Losses on a three-phase transmission line is dependent on the phase resistance R and phase current I and can be written according to equation (47)

$$P_l = 3 \cdot R \cdot I^2 \quad (47)$$

where the phase current I can be written according to equation (48)

$$I^2 = \bar{I} \cdot \bar{I}^* = \frac{\bar{S}}{\sqrt{3} \cdot \bar{U}} \frac{\bar{S}^*}{\sqrt{3} \cdot \bar{U}^*} = \frac{S^2}{3 \cdot U^2} = \frac{P^2 + Q^2}{3 \cdot U^2} \quad (48)$$

where \bar{S}^* and $\bar{I} \cdot \bar{I}^*$ are the complex conjugate respectively. The transmission line losses in equation (47) can now be written as described in equation (49)

$$P_l = R_{kj} \frac{P_{kj}^2 + (Q_{kj}^2 + bU_k^2)^2}{U_k^2} \quad (49)$$

The index kj refers to "from bus k to bus j " and bU_k^2 refers to the reactive power produced in bus k by the transmission capacitance.

Correspondingly, the reactive power transmission losses in equation (50) are

$$Q_l = 3 \cdot X \cdot I^2 = X_{kj} \frac{P_{kj}^2 + (Q_{kj}^2 + bU_k^2)^2}{U_k^2} \quad (50)$$

As seen in equations (47) and (50), an increase in voltage would decrease the losses, thus higher transmission voltage leads to smaller losses. Additionally, transmission of reactive power increases the losses, which is why it is desirable to produce reactive power locally. [13]

III. MATPOWER

MATPOWER is a simulation tool constructed as a package of m-files used for solving power flow problems in MATLAB [15], and is used in this project. MATPOWER was motivated by its ability to efficiently solve problems associated with this project, while at the same time being relatively easy to understand. The steady state power flow problem is solved with Newton-Raphson's method. Two types of files are needed to run the simulations. The first type, *case files*, are MATLAB-structs that specifies the analyzed case by defining and assigning values to the components in the grid.

First, the base value for the system MVA, \bar{S}_{base} in equation (20) is used to convert powers and voltages into per-unit quantities.

Second, the buses, such as active and reactive power consumption, are set. Here the bus classifications are determined by defining them as PQ-buses, PU-buses or slack-buses [16]. In this project the cities and wind parks are PQ-buses since the active and reactive power are known. The hydropower plants are PU-buses since the active powers and voltage magnitudes are known.

Third, the slack-bus is determined as a reference bus with the voltage magnitude and angle known.

Fourth, the transmission lines, resistance, reactance, and capacitance per-unit values are assigned. MATPOWER uses the Π -model for the branches [16].

The second type, the *main file*, is where the user inputs data to use in the case file. The calculations required to solve the power flow system are done here as well. In this project a time-series is handled in the main file. The data for the time series is defined for each hour and includes the power consumption data for the cities and the production data for the power plants. This data is injected iteratively in a for-loop into the case-file for each hour and a power flow calculation is carried out every iteration. The results from this calculation in the form of voltages and power flows are extracted every iteration and stored in matrices.

IV. CASE STUDY

The area of study is in Jämtland county, Sweden, which is a part of the electricity trading area SE2. It is conducted as a green field study, taking into account nature reserves. The original power system consists of five hydropower plants, five loads and one slack bus that is a fully functional system with acceptable voltage levels. A sub-goal is to introduce wind and solar into the system and investigate if the voltage levels can be kept within $\pm 10\%$. Another goal is to examine if the reactive power produced by the solar power could reduce the transmission losses and increase the power factor the same way shunt capacitors can, but with the benefit of having active power production when the sun shines. This chapter is divided into sub-sections where each part of the power system is described accordingly.

A. Loads

The loads consists of five cities, as shown in Table I, and they are modeled as PQ-buses in the simulations. The data is based on hourly consumption data for 2020, for each load. The data for a load is obtained by multiplying the total consumption of trading area SE for a specific hour by the share of yearly average [17] for a load and an estimated SE2 total. Since SE2 is larger than the area of study, using data for the whole SE2 would not be accurate. Therefore, two large loads, Skellefteå and Gävle was excluded for better accuracy.

TABLE I
LOAD DATA

#	Load	Consumption [GWh/year]
1	Åre	228
2	Krokom	162
3	Östersund	539
4	Bräcke	83
5	Strömsund	131

B. Hydropower

Hydropower is one of the oldest and most reliable renewable power sources. Due to geographical circumstances Sweden has a relatively large and well established hydropower expansion and transmission line development, with rivers up north for production and big cities in the south for consumption [18]. In this project, five hydropower plants are part of the basic system, see Table II. The hydropower plants are modeled as PU-buses in the simulations.

TABLE II
HYDROPOWER PLANT DATA

#	Hydropower Plant	Capacity [MW]
6	Torrön	24
7	Mörsil	40
8	Kvarnfallet	19
9	Stugun	41
10	Svarthålsforsen	80

C. Wind Power

One of the most important aspect regarding wind power plant locations is average wind speeds. Nature reserves and other landscape protection areas are considered, in the final decision making, see Fig. 8. High wind speed close to cities is also disregarded as it is not conceivable to build wind power that close to cities. A wind turbine, Vestas V90 2000, is chosen for being commonly used in the industry [19]. There are three wind parks introduced in this project, Knutkaribränna (100 MW), Millestbodarna (150 MW) and Norder-Rensjön (200 MW). They are fictive wind parks, but the installed power data is based on actual wind parks in the region [20] [21]. In the simulations the wind parks are modeled as PQ-buses. As shown in Table III, a share of total load is also displayed as a reference of how big the wind parks are. The share is calculated by multiplying the power installed with an average of 3000 hours of maximum power over a year [22].

D. Solar Power

Solar panels are typically placed on rooftops [24] on houses and apartment complex. Solar panels can not produce reactive

TABLE III
WIND POWER PARK DATA

#	Wind Power Plant	Capacity [MW]	Share of Total Load [%]
11	Norder-Rensjön	200	47
12	Millestbodarna	150	35
13	Knutkaribränna	100	23

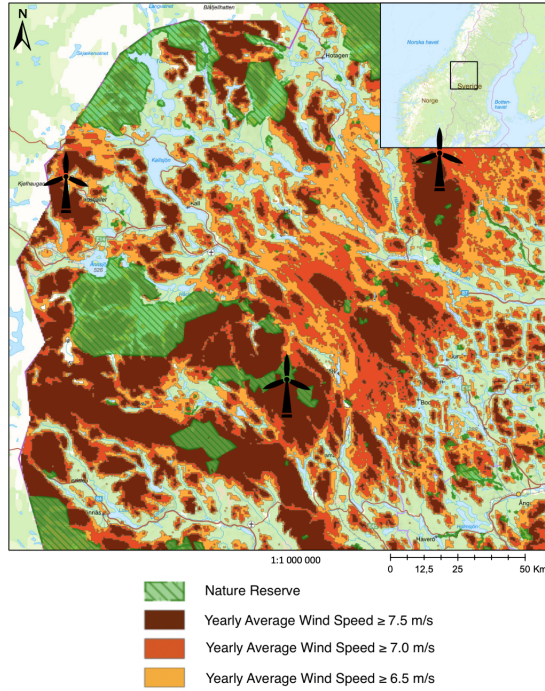


Fig. 8. Yearly average wind speeds and area restrictions [23].

power on its own, since it produces DC. Although, converting it to AC using a power converter makes it possible for reactive power to be drawn, depending on the load [25]. Solar power could therefore produce reactive power to help balance the load. [26]. The solar power was installed in the cities and is shown in Table IV.

TABLE IV
SOLAR POWER DATA

City	Power Installed Per County [MW]
Strömsund	0.65
Bräcke	0.73
Åre	2.38
Krokom	3.79
Östersund	10.1

E. Transmission Lines

Two different types of transmission line connections are considered, meshed and radial. The meshed grid is shown in Fig. 9. The radial grid is shown in Fig. 10. Both grids have a transmission voltage of 220 kV. The most common transmission voltage in Sweden is 400 kV [27]. Although, given the size of the power system in this project, the 220 kV is more suitable. The slack bus (number 14) is used in the simulations as a representation of a connection to the larger national grid. This bus balances the system by importing or exporting power as necessary.

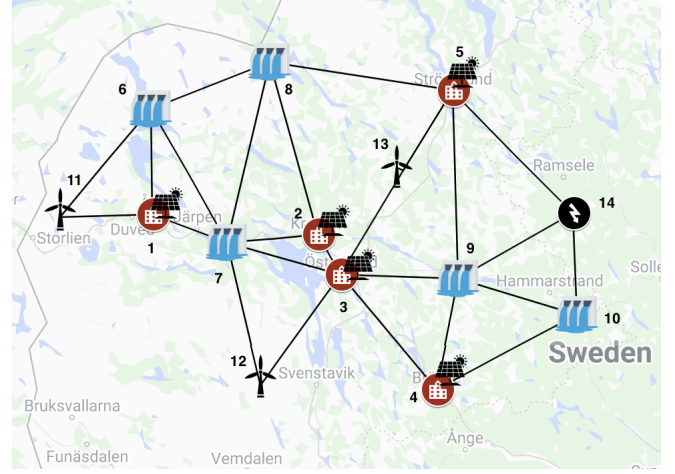


Fig. 9. Meshed power system [28].

The radial connection is a simpler and more cost-effective connection type since the amount of additional transmission lines needed is reduced, see Fig. 10.

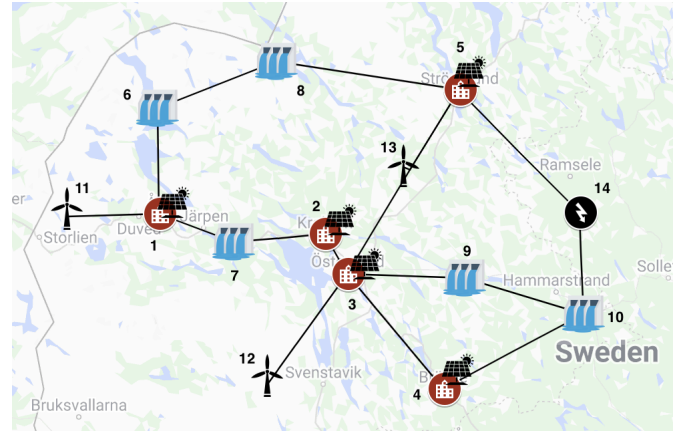


Fig. 10. Radial power system [28]

F. Grid Codes

A grid code is a directive to determine the requirements to maintain grid stability and can be used as a planning document and support for decision guidance. An example of how a specific grid code requirement is implemented in this project regarding the reactive power can be seen in Fig. 11. It shows a variable displacement factor that is dependent on the active power, $\cos \varphi(P)$ [29].

G. Data Simulation Period of Interest

Three different weeks of 2020 with different scenarios is considered, which can be seen in Table V. The first week of interest is when there is maximum wind power production, the second week is when there is maximum load in the system, the third week is when there is maximum solar production and the fourth week is when there is high wind production in combination with low load. The solar data is collected by

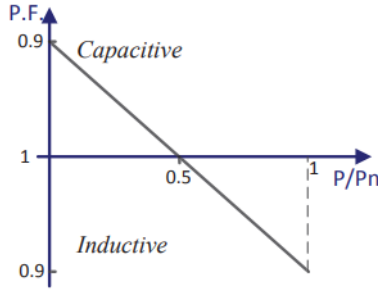


Fig. 11. An example of a variable displacement factor depending on the active power, $\cos \varphi(P)$ [30].

comparing the mean values of different weeks with high peaks of solar production to ensure that the maximum production week is obtained. Looking only at the highest peaks could result in a lower weekly average and thus not give the week of maximum solar production.

TABLE V
WEEKS OF INTEREST (2020)

Attribute	Week	Date
Maximum wind power	3	January 13 - 19
Maximum load consumption	9	February 24 - March 1
Maximum solar power	23	June 1 - 7
High wind + low load	38	September 14 - 20

H. Reactive Power Compensation

Since the time resolution of the simulations is one hour, shunt capacitors and reactors were used in this report. Shunt reactors were installed in each of the hydropower plants in order to compensate for the surplus of reactive power in the system. Without the reactors, the reactive power consumption levels in the hydropower plants were unrealistically high. The size of the reactor is measured in MVar and was set to the level of the reactive power consumption of the power plant when the reactors were not installed. This was done with the goal of keeping the size of the reactive power production/consumption in each hydropower plant less than 5% of the active power production.

I. Cases

Five cases are created in total. Three base cases with installed hydro, wind and solar power based on real world values in the Jämtland area are referred to as *Present cases*. The difference between the *Present cases* is the types of power production installed. In the *Hydro* case only hydropower is included, in the *Present Wind* case the wind power is added and in the *Present Solar* case the solar power is added.

Two cases are created in order to illustrate an increased amount of wind and solar power, and are referred to as *Increased cases*. The *Increased Wind* case doubles the amount

of wind power in each wind park. The *Increased Sun* case quadruples the amount of solar power in each city while keeping the wind power at the same level as in the present case.

Every case is simulated with a mesh transmission line structure as shown in Fig. 9 and a radial transmission line structure as shown in Fig. 10.

V. RESULTS

In this section the results from the simulations are presented. First the network structure effects and voltage stability are evaluated. Then the study of reactive power compensation, followed by solar power as a reactive power compensator.

A. Transmission Line Structure and Voltage Stability

A difference between the mesh transmission line system and the radial transmission line system is the amount of shunt reactors necessary. As the radial system has a shorter total transmission line length the reactive power produced in the transmission lines is smaller. This results in a smaller need for shunt reactors consuming reactive power. The reactive power levels in the meshed and the radial system exhibit similar behaviors in the simulations, meaning that peaks and valleys occur at the same time in both of the systems with slight differences in the magnitudes. For this reason and for easier comparison, only results from the radial system are presented later in the sections treating reactive power.

Another difference is the voltage magnitude sensitivity. In the radial system, increasing the amount of intermittent renewable energy such as wind and solar affects the voltage magnitude to a greater extent than in the mesh system. This can be exemplified by looking at the voltage magnitudes for both transmission line structures during week 38 with the wind power connected. In Fig. 12 and 13 the voltage magnitudes are shown and in Fig. 14 the injected active power from the wind power is shown. There is a stronger correlation between high wind power production and lowered voltage levels in the radial system than in the mesh system. The deviations in the radial system are still small, but bigger than in the mesh system.

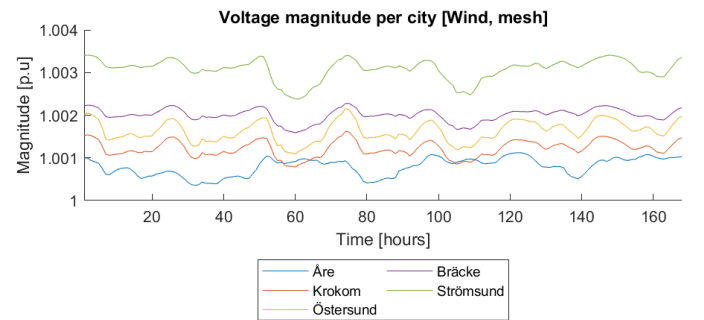


Fig. 12. Voltage magnitudes in each city in the meshed present wind case during week 38, showing slight voltage deviations during hours 60 and 110 when the wind power production increases.

The radial system therefore risks breaking the set voltage limits before the mesh system when increasing the installed wind power. The voltage magnitude in Strömsund for the same

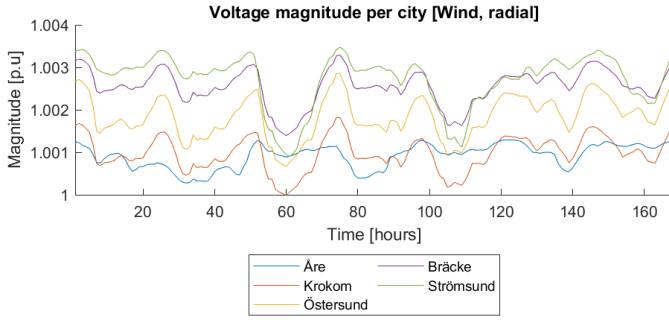


Fig. 13. Voltage magnitudes in each city in the radial present wind case during week 38, showing greater voltage deviations during hours 60 and 110 when the wind power production increases than in the meshed system.

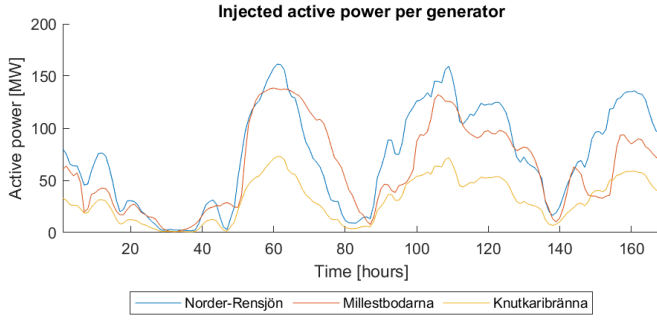


Fig. 14. Wind power production for the present wind case during week 38 with peak production during hours 60 and 110.

week as before but with double the installed wind power is shown in Fig. 15, where the radial system deviates more than the meshed.

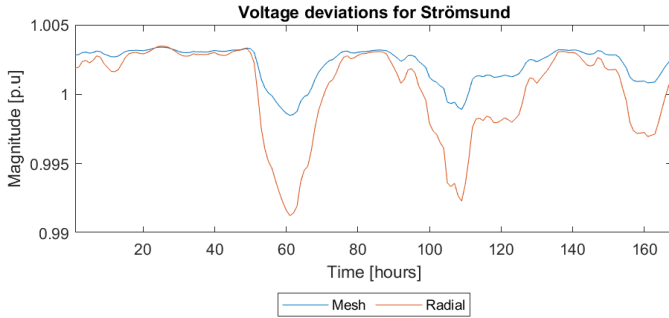


Fig. 15. Voltage deviation difference in Strömsund between the mesh and radial transmission line structure for the increased wind case. A bigger deviation occurs in the radial system when the wind power production is active.

Increasing the amount of power production in the hydropower plants does not give bigger voltage deviations in the radial system, than in the meshed system. This is due to the hydropower plants being modeled as PU-buses where the voltage magnitude is set to the reference value of 1 p.u.

The voltages in each of the present cases were kept within the set level of $\pm 10\%$ of the reference value. The biggest voltage deviation in the present cases occurred in the radial transmission line structure case with wind power installed. This deviation of $+0.36\%$ happened in Strömsund during week 3 on hour 124 as shown in Fig. 16.

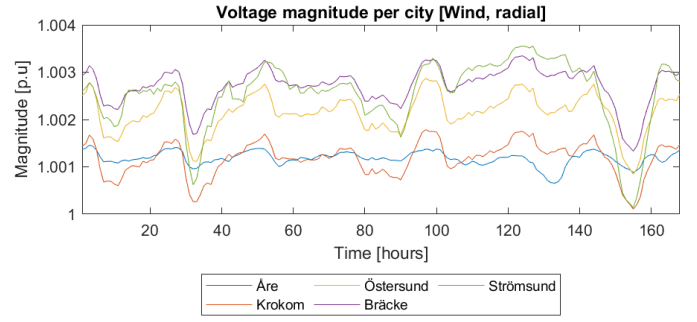


Fig. 16. Voltage magnitudes in each city during week 3 in the present case. The greatest deviation in the present cases occur during hour 124 in Strömsund.

In the increased cases the largest voltage deviation of -1.38% occurred in the radial wind system during week 3 on hour 155 as shown in Fig. 17. This deviation correlates with the wind power production in Fig. 14 (half the amount in the present case) in both the present and the increased case.

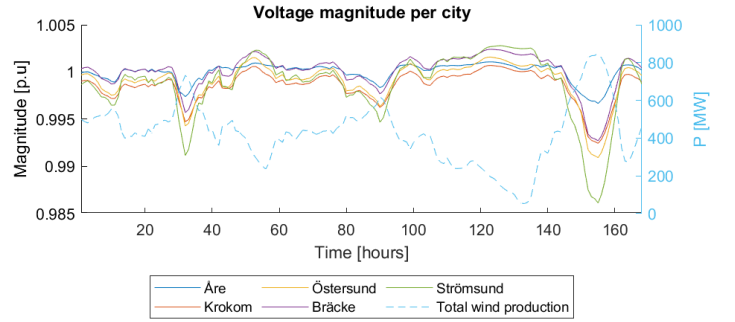


Fig. 17. Voltage magnitudes in each city (left y-axis) and wind power production during week 3 (right y-axis) in the increased case. The biggest deviation occurs during hour 155 in Strömsund, when the wind power production is the highest.

B. Reactive Power Compensation

The reactive power levels in the system needed to be compensated because of the excessive reactive power production in the transmission lines. The goal of keeping the size of the reactive power production/consumption in each hydropower plant less than 5% of the active power production was not fulfilled. The encountered problem can be generally illustrated by looking at the present hydro case with the radial transmission line structure. The fraction of reactive power for *Stugun* is shown in Fig. 18, where the reactive power injection of the power plant is divided by the active power injection. It shows that the hydropower plant breaks the limit of 5% both when consuming and producing reactive power. This problem is not solvable by installing bigger or smaller shunt reactors in the hydropower bus, as a bigger reactor simply shifts the curve up and a smaller reactor shifts the curve down.

Introducing the wind parks further nuances the reactive power compensation needs. Looking at the same week and network structure as before, but now with wind power introduced, in Fig. 19 active power surges from the wind power parks shown in Fig. 14 result in reactive power spikes in the

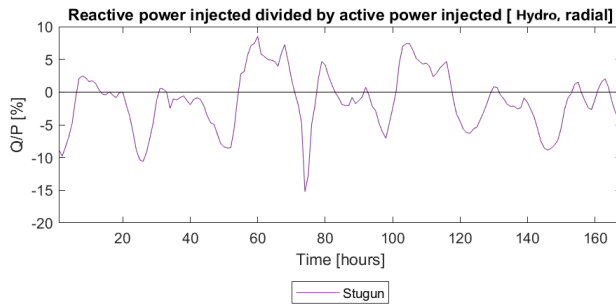


Fig. 18. Fraction of reactive power for Stugun in the present hydro case during week 38. The reactive power level both exceed and fall below the limit of $\pm 5\%$.

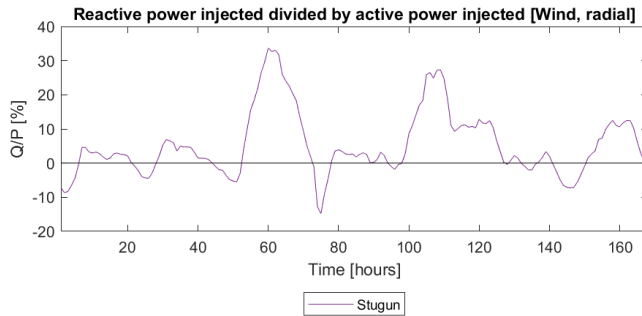


Fig. 19. Fraction of reactive power for Stugun in the present wind case during week 38. The reactive power level deviates more from $\pm 5\%$ when wind power is introduced.

hydropower plants. This is an issue that cannot be resolved by adjusting the size of the installed shunt reactors.

Increasing the amount of installed wind power results in higher reactive power losses in the system. This decreases the need for shunt reactors and possibly creates a need for shunt capacitors in order to compensate for the reactive power losses, depending on how much wind power is installed. This is shown in Fig. 20, where the installed wind capacity is doubled and the reactive power injections from the hydropower plants have increased.

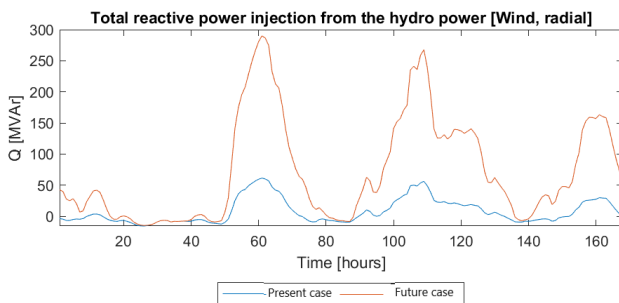


Fig. 20. Reactive power injection of the hydropower plants during week 38, showing that an increase in wind power production brings an increase in reactive power production.

C. Solar Power as a Reactive Power Compensator

The use of solar power in the cities has a small effect on the reactive power levels and voltage magnitudes, and the results from the simulations are almost identical to the simulations

with wind power. This is because of the small amount of power contributed from the solar power. Fig. 21 and 22 show the active and reactive power injections from the solar power systems for each week. The total installed solar power is 17.6 MW which is smaller than the smallest hydropower plant *Kvarnfallet* in the system.

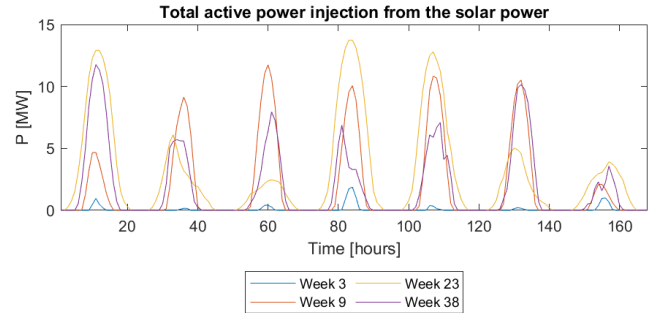


Fig. 21. The total active power injection of the solar power systems for each week.

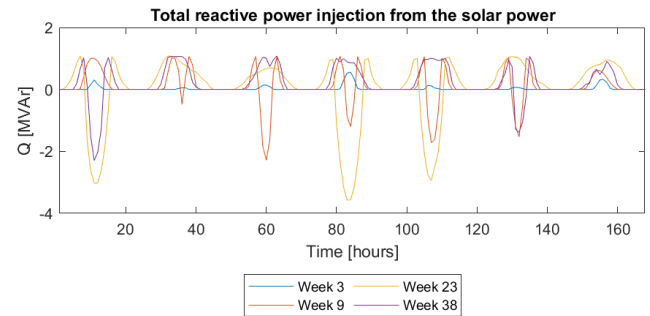


Fig. 22. The total reactive power injection of the solar power systems for each week.

Increasing the installed solar power yields a greater effect on the active power and voltage. Week 23 is the week with the highest solar irradiance as this is a week in the beginning of the summer. This results in the solar power having the greatest impact on the reactive power levels and the voltage magnitudes during this week. The effects of increasing the installed power can be studied by first looking at the radial system during week 23. In Fig. 23 and 24 the voltage magnitudes and the reactive power levels are shown for the radial present wind case during week 23.

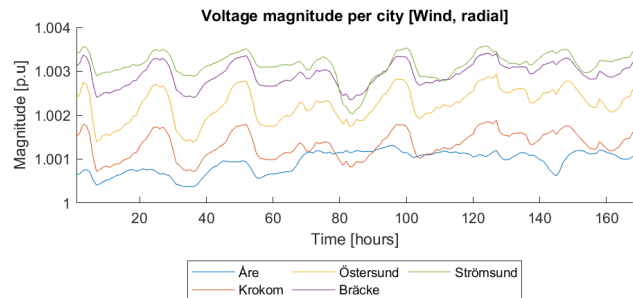


Fig. 23. Voltage magnitudes for the cities in the radial wind system during week 23, without solar power connected.

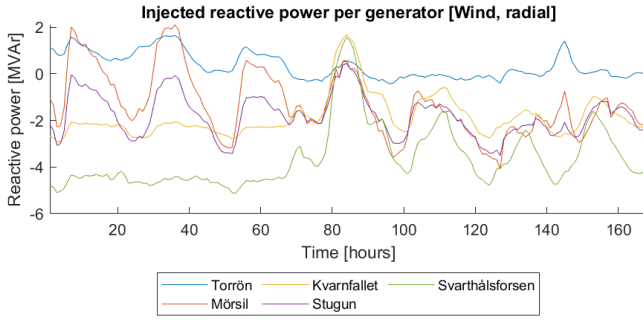


Fig. 24. Reactive power injections of the hydropower plants in the radial wind system during week 23, without solar power connected.

Then the solar power is introduced, this time four times larger than the solar power installed before. The voltage magnitudes and reactive power levels for the radial increased solar case are shown in Fig. 25 and 26 respectively. A connection between the reactive power injections from the hydropower plants and the voltage magnitudes is shown. The reactive power and voltage magnitude spikes in Fig. 26 and 25 respectively, correlate to the reactive power consumption in the solar power systems in Fig. 22. The fraction of reactive power for the wind case and the solar case respectively are shown in Fig. 27. It shows that the solar power contributes with larger reactive power fraction deviations from the goal of 5%.

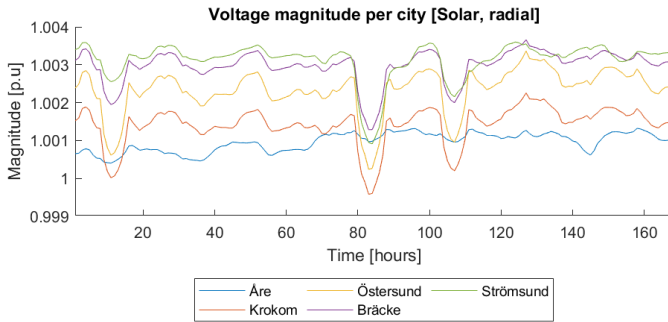


Fig. 25. Voltage magnitudes for the cities in the radial solar system during week 23 with four times the installed solar power. The deviations correlate to the reactive power consumption in the solar power systems in Fig. 22

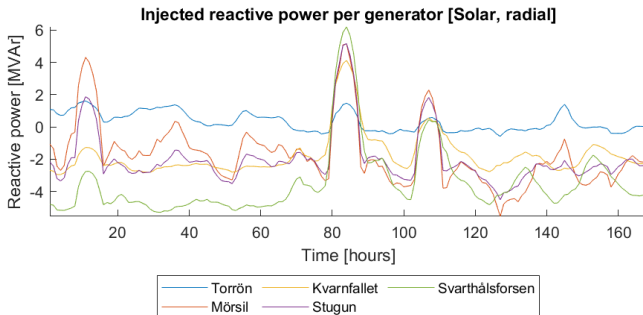


Fig. 26. Reactive power injections of the hydropower plants in the radial solar system during week 23 with four times the installed solar power. The deviations correlate to the reactive power consumption in the solar power systems.

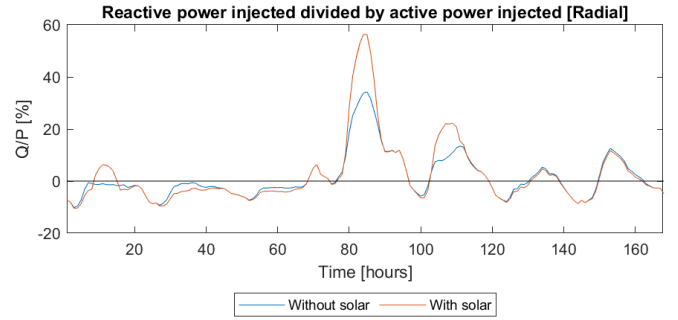


Fig. 27. Reactive power injections of the hydropower plants in the radial system without and with solar power installed during week 23. Using solar power as a reactive power compensator has the opposite effect of the desired one, as the deviation from $\pm 5\%$ is greater with solar power.

VI. DISCUSSION

A. Project Limitations

The simulated power system is not a representation of a real power system in the real world. It is a small system with only one voltage level, a limited amount of buses and it lacks some parts of the interaction with the bigger, national grid that it is a part of. Interaction with the national grid is limited to power import and export as seen fit for the simulated system, and does not take into account the national power demand, power prices or system limitations in the bigger grid. This affects the results in such a manner that they do not accurately represent the power system in Jämtland, but does not disqualify the results of this study as the aim is to illustrate how a generalized fictive power system behaves. The transmission lines are drawn as straight lines between the buses with no power transmission limit. Not all hydropower plants in the area have been included in the system and not all cities either. The wind parks are placed with only yearly average wind conditions and nature reserves as parameters. The power system is a socio-technical system but almost exclusively technical aspects are considered. Social values such as the native peoples reindeer pasture claims and cultural heritage areas are not considered in this study when placing wind parks and transmission lines. Administrative restrictions in the form of policies are excluded as well. In order to create a complete picture of the power system in Jämtland, the scope of the study needs to be widened to include aspects such as these.

B. Voltage Stability

The voltage levels are affected by renewable intermittent power production as shown in Fig. 17. However, the biggest voltage deviation of -1.38% is far below the set limit of $\pm 10\%$. This shows that the limiting factor when introducing solar and wind power in this system is not the voltage levels. The main obstacle for this is the reactive power levels in the system. That the voltages are as steady as they are, is likely because of the high amount of hydropower in relation to the load centers. The hydropower are PU-buses with a set voltage magnitude of 1 p.u and the high PU-bus ratio helps maintain the voltage levels in the system since every load center is connected to a hydropower plant. Modeling the hydropower

plants as PQ-buses would result in wanted reactive power levels as the active and reactive power production in the power plants would be predetermined. However, this comes with the trade-off of increased voltage instability in the system as the voltage magnitude would no longer be set to 1 p.u in the hydropower buses. Thus the limiting factor when introducing solar and wind power is decided by the system modeling method.

The results show that the voltage levels are affected to a greater extent when the amount of wind power production is increased. This shows that the system will not be robust enough to keep the voltage deviations within the set limit of $\pm 10\%$ if too much wind power is introduced.

C. Reactive Power

The results of this study show that reactive power compensation is vital for keeping the reactive power levels within acceptable limits. It shows what sort of compensation is necessary, but do not give information on what components should be used or how to implement them in the system.

By looking at Fig. 18 it is clear that even in the system without renewable energy introduced, there is a need for accurate and variable reactive power compensation. This is something that cannot be achieved with only passive shunt reactors and capacitors, and requires active and continuous compensation systems with a fast time response as is brought up in [8].

The need for these sorts of reactive power compensation systems is further reinforced when wind power is connected to the simulated power system. Reactive power production/consumption peaks are what cannot be handled by passive shunt reactors and capacitors. These peaks increase in size and possibly number depending on the reactive power levels before wind is introduced as shown in Fig. 19 when wind power parks are connected. This is because of the intermittent nature of the power production with the consequence of power production peaks and valleys. An explanation for an increased amount of wind power leading to an increased production of reactive power in the hydropower plants can be found in the fact that the increased active power levels bring with them higher reactive losses, which the hydropower plants compensates for. This relation is described in equation (50).

Using the solar power as a potential reactive power compensator is shown in theory in this project, since a connection between solar irradiance and reactive power levels is shown. However, the solar power system do not provide the desired compensation effects in the simulated base system. The installed solar power was not big enough to have an impact on the reactive power levels. Increasing the installed solar power resulted in an impact on the system, but an opposite to the desired one which was to keep the reactive power levels within the limit. This is shown in Fig. 27 where the reactive power deviation from the set limit of 5% is greater when solar power is added. This shows that the implemented grid code does not produce desired results in the simulated system. A grid code based on the current reactive power in the system would probably produce a better result, but would still have the limitation of use only when the sun is shining.

D. Grid Structure

This study provides insight in some general effects of the transmission line structure for the simulated system, but are not conclusive enough to show a clear connection to the voltage stability of all power systems. However, there are indications that the voltage levels deviate more in a radial system than a mesh system when introducing wind and solar power. This does not mean that a mesh system is needed when connecting wind and solar, but grid reinforcements could be necessary in order to maintain voltage stability. If additional transmission lines are constructed in order to reinforce the grid, larger reactive compensation systems might be needed. This may be necessary in order to compensate for the increased line charge leading to a larger amount of reactive power being produced in the transmission lines.

E. Future Development Strategies

Research into how to better manage the reactive power levels with better reactive power compensation will enable a better analysis on how to introduce renewable intermittent power sources. What grid codes that brings about the best compensation is also an interesting study area. The results also indicate that reinforcements need to be made in the grid when variable power production is connected, and therefore research into this could be of interest.

VII. CONCLUSION

A fictive power system has been simulated with the use of MATPOWER and MATLAB in combination with historical hourly data of power production and consumption, wind speeds and solar irradiance. The power flow simulations provided insight into how the voltage magnitude and power levels behave. The voltage levels in the simulated system are kept within the set limits of $\pm 10\%$, with the largest deviation being -1.38% . However, the reactive power levels are not kept within the set limits of 5% of the active power levels in the hydropower plants. This is because of the lack of variable and continuous reactive power compensation. The need for this kind of compensation is increased when variable power sources such as wind is introduced to the grid. Solar power systems have the potential to provide reactive power compensation, but in the simulated system it does not have the desired compensation effect. Increasing the installed solar power only result in the opposite of keeping the reactive power levels within the limit. For it to have better results, the size of the installed solar power needs to be considered along with the grid code used. Transmission line structure seems to have an effect. There is a stronger correlation between high wind production and lowered voltage levels in the radial system than the mesh system. This implies that when introducing variable renewable power sources, reinforcements in the grid may be needed.

ACKNOWLEDGMENT

The authors would like to thank Evelin Blom and Lennart Söder for their help and guidance. A special warm thank you

to Evelin for her tremendous support, invaluable feedback and insight. Always taking the time to answer questions and offer solutions.

REFERENCES

- [1] WorldEnergyCouncil. (2019, Apr) World energy insight brief 2019 - global energy scenarios comparison review. [Online]. Available: <https://www.worldenergy.org/assets/downloads/WEInsights-Brief-Global-Energy-Scenarios-Comparison-Review-R02.pdf>
- [2] UnitedNationsFrameworkConventiononClimateChange. (2021, Mar) A brief guide to renewables. [Online]. Available: <https://unfccc.int/blog/a-brief-guide-to-renewables>
- [3] B. N. Stram. (2016, Sep) Key challenges to expanding renewable energy. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0301421516302646>
- [4] N. Grid. (2018, Jun) Voltage and frequency dependency. [Online]. Available: <https://www.nationalgrideso.com/sites/eso/files/documents/SOF\%20Report\%20-\%20Frequency\%20and\%20Voltage\%20assessment.pdf>
- [5] N. Dyess. (2018, Apr) Sags, swells, and voltage deviation effects for troubleshooting motors. [Online]. Available: <https://www.plantengineering.com/articles/sags-swells-and-voltage-deviation-effects-for-troubleshooting-motors/>
- [6] M. Iorgulescu and D. Ursu. (2017, Apr) Reactive power control and voltage stability in power systems. Cham. [Online]. Available: https://doi.org/10.1007/978-3-319-51118-4_6
- [7] B. Kirby and E. Hirst, "Ancillary service details: Voltage control," Dec 1997.
- [8] X. Qiao, J. Bian, C. Chen, and H. Li, "Comparison and analysis of reactive power compensation strategy in power system," in *2019 IEEE Sustainable Power and Energy Conference (iSPEC)*, 2019, pp. 689–692.
- [9] H.-P. Nee, M. Leksell, S. Östlund, and L. Söder, *Eleffektsystem*. KTH: Stockholm, 2019.
- [10] E. M. Malatji and B. Chabangu, "Innovative method for power factor correction using a solar plant as a source of reactive power," in *2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC)*, Dec 2018, pp. 1–5.
- [11] L. Söder. (2005, Jan) Statisk analys av elsystem. KTH. [Online]. Available: <http://www.diva-portal.org/smash/get/diva2:1075656/FULLTEXT01.pdf>
- [12] W. Huang and D. J. Hill, "Network-based analysis of long-term voltage stability considering loads with recovery dynamics," *International Journal of Electrical Power & Energy Systems*, vol. 119, p. 105891, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0142061519328297>
- [13] L. Söder and M. Ghandhari. (2015) Static analysis of power systems. KTH. [Online]. Available: https://www.kth.se/social/files/55f17d7df2765458ad9c151b/comp_EG2100_HT15_v2.pdf
- [14] H. R. Pota, *The Essentials of Power System Dynamics and Control*, 1st ed. Singapore: Springer Singapore, 2018.
- [15] MATPOWER. (2022, Apr) About matpower. [Online]. Available: <https://matpower.org/about/>
- [16] R. D. Zimmerman and C. E. Murillo-Sánchez, "Matpower user's manual," Oct. 2020. [Online]. Available: <https://doi.org/10.5281/zenodo.4074122>
- [17] Statistikdatabasen. (2022, Apr) Energidata (mwh) efter län och kommun, kategori samt energityp. År 1990 - 2008. [Online]. Available: https://www.statistikdatabasen.scb.se/pxweb/sv/ssd/START__EN__EN0203/EnergiKommKat/
- [18] Vattenfall. (2022, Apr) The revolution of electricity - before vattenfall. [Online]. Available: <https://history.vattenfall.com/stories/the-revolution-of-electricity/before-vattenfall>
- [19] Vestas. (2021, Dec) Track record brochure. [Online]. Available: <https://nozebra.ipapercms.dk/Vestas/Communication/Trackrecord/track-record-q1-2021/?page=2>
- [20] Statkraft. (2022, Apr) Mörttjärnberget vindpark. [Online]. Available: <https://www.statkraft.se/om-statkraft/alla-vara-anlaggningar/sverige/morttjarnberget-vindpark/>
- [21] VasaVind. (2020, Apr) Vasa vind driftsätter Åskälen vindkraftpark. [Online]. Available: <https://www.vasavind.se/nyheter/pressmeddelande/vasa-vind-driftsatter-askalen-vindkraftpark>
- [22] VästraGöralandsregionen. (2017, Feb) Så mycket energi producerar ett vindkraftverk. [Online]. Available: https://www.vgregion.se/regional-utveckling/verksamhetsomraden/miljo/power-vast/fakta-om-vindkraft/energi--teknik/?vgrform=1#Sa_mycket
- [23] Länsstyrelsen. (2022, May) Vindbrukskollen. [Online]. Available: <https://vbk.lansstyrelsen.se/>
- [24] SolarEngineeringGroup. (2022, Mar) The best location for solar panels. [Online]. Available: <https://www.sunengis.com/the-best-location-for-solar-panels/>
- [25] A. Hassan, A. Dash, and D. De, "Comparison of converter structures for residential pv system with module based maximum power point tracking," in *2018 Technologies for Smart-City Energy Security and Power (ICSESP)*, 2018, pp. 1–6.
- [26] GoHz. (2016, Aug) Does solar panel generate reactive power? [Online]. Available: <http://www.gohz.com/does-solar-panel-generate-reactive-power>
- [27] SvenskaKraftnät. (2021, Feb) Teknik. [Online]. Available: <https://www.svk.se/om-kraftsystemet/om-transmissionsnatet/teknik/>
- [28] GoogleMyMaps. (2022, May) G1 voltage meshed. [Online]. Available: <https://www.google.com/maps/d/edit?mid=1YM3oCsTaGdr5gFGqcE0tWbWE9w974PpK&ll=63.154678276162656\%2C13.96349179843752&z=8>
- [29] E. Troester, "New German grid codes for connecting pv systems to the medium voltage power grid," in *2nd International workshop on concentrating photovoltaic power plants: optical design, production, grid connection*, 2009, pp. 1–4.
- [30] A. Samadi, M. Ghandhari, and L. Söder, "Reactive power dynamic assessment of a pv system in a distribution grid," *Energy procedia*, vol. 20, pp. 98–107, 2012.

Capacity Market in the US

Astrid González and Tuve Gustavsson

Abstract—As the transition from the use of fossil fuels to renewable sources takes place, consumption of electricity will greatly increase. The shift in energy sources will have a deep impact on how financial decisions in the grid will be made. To ensure that the necessary investments are made to meet the new needs, many network administrators have used different types of markets for capacity. This project reviews how the network administrator PJM in the US uses capacity markets to secure the supply of electricity and stability in the grid. A literature study was conducted together with a market simulation and the results from the simulation shows that the use of a separate capacity market is a successful concept for securing future electricity supply and stability in the grid.

Sammanfattning—I takt med att övergången från användningen av fossila bränslen till förnybara källor sker, kommer elförbrukningen att öka kraftigt. Skiftet av energikällor kommer att ha en stor påverkan på hur finansiella beslut inom elnätet kommer att tas. För att se till att nödvändiga investeringar görs för att klara de nya behoven har många nätverksadministratörer använt sig av olika slags marknader för kapacitet. I detta projekt undersöks hur nätverksadministratören PJM i USA använder sig av kapacitetsmarknader för att säkra elproduktionen och stabiliteten i nätet. En litteraturstudie genomfördes tillsammans med en marknadssimulering och resultatet från undersökningen visar på att användningen av en separat kapacitetsmarknad är ett framgångsrikt koncept för att säkra framtida elförsörjning och stabilitet i nätet.

Index Terms—Forward capacity market, Variable Resource Reserve, Installed Reserve Margin, Supply curve, Demand curve, Cost of New Entry

Supervisors: *Mohammad Reza Hesamzadeh*

TRITA number: *TRITA-EECS-EX-2022:145*

ACRONYMS

PJM	Pennsylvania, New Jersey and Maryland
E&AS	Energy & Ancillary Services
CONE	Cost of New Entry
EFORD	Equivalent Demand Forced Outage Rate
IRM	Installed Reserve Margin
RelReq	Region Reliability Requirement
VOLL	Value of Lost Load
LOLP	Loss of Load Probability
RTO	Regional Transmission Organization
NUC	Nuclear resource
COA	Coal resource
CYC	Combined Cycle
TUR	Combustion Turbine
TRA	Solar PV (Tracking)
FIX	Solar PV (Fixed)
ONS	Onshore Wind
OFF	Offshore Wind
BAT	Battery Storage

I. INTRODUCTION

The recent debates on climate change have sparked a discussion on how to best achieve a transition to clean energy production. As discussed in [1], the Paris agreement shows a political will to reduce global warming to a maximum of 2 degrees Celsius. To achieve this, electricity in the form of renewable sources has been proposed as a solution.

However, these sources are dependent on the weather for their production; if the wind does not blow, we can not produce any wind power. In situations of low production and high consumption, the electricity prices will go up. In a perfect market, this rise in costs would be enough for some consumers to voluntarily reduce their consumption.

However, as pointed out in [2], in many countries the government has enforced a price ceiling to prevent prices from reaching above levels that would be politically unstable. This kind of market intervention will create a “Missing Money Problem”, since investors will miss out on potential profit. Furthermore, the typical consumer is not actively engaging in the electricity market and will consume power regardless of the current price.

Together, these factors make the otherwise perfect market unable to self-regulate, and we will need to plan for periods of generation scarcity through capacity reserves. If no action is taken, we will eventually face a blackout. In a simplified way, a capacity reserve is a reserve of production that can be available on short notice.

At the same time, the procurement of capacity is not an easy task. In the traditional electricity market (which is called an energy-only market), the generation owner only gets paid for delivered energy, not for being on stand-by. Furthermore, as is discussed by [3], the penetration of renewable sources is lowering the market prices, which makes it harder for investors to cover their capital costs. These factors create uncertainty for investors about whether or not an investment would turn out profitable. As we can see, it is highly motivated to investigate how to create incentives for investments in new generation capacity through market mechanisms.

Today there exist a number of different strategies to combat the problem with new investments. Like [2] mentioned, one possible solution is to separate the energy-only market from a capacity market. In contrast to the energy-only market, the capacity market will pay generation owners for their availability. The owners are being paid regardless of whether their generation is needed or not. Another option proposed by [4] is to subsidize generators instead. On the other hand, there are those who advocate that a better solution would be to improve the pricing system during times of scarcity.

A. Objectives

This paper aims to review whether or not the use of a forward capacity market is an effective method for procuring enough capacity in the system. This will be done by studying how one of the regional transmission organizations in the US, PJM, has implemented it in its market. Therefore, this paper seeks to answer the following question:

- Are capacity markets an effective way to procure desired generation capacity levels?

This paper defines desired generation capacity levels as the level of capacity needed to expect one blackout every ten years. However, several studies such as [5] and [6] has used definitions from social-welfare theory. However, this is out of scope for this paper, but a paragraph under section V about the social-welfare perspective will be included.

B. Struture

This paper will first provide the reader with a theory section. The theory will then be put into use under section III with the use of simulations. In section IV, the results from the simulations will be presented, and the paper will end with a discussion about the results.

II. THEORY

A. Adequacy problem

As a consumer, we expect that whenever we turn on our light switch, our lamp will be turned on instantaneously without any delay. It would be outrageous to be informed that we would have to wait a couple of minutes for new electricity to be produced to meet our needs. This is what resource adequacy is all about: being able to meet the consumers need for energy.

However, when electricity becomes scarce, the prices will rise. Without any administrative actions, prices during periods of high consumption and low production would increase beyond what would be considered politically acceptable, which has lead to many countries enforcing a price-cap on the market.

When introducing a price-cap on prices, investors will naturally have lower incentives to invest in generation capacity since there is now much less money to earn, and the investments might not be profitable enough. As explained in the introduction, this is what is call the "Missing Money Problem".

However, if we would raise the offer-cap, there would be difficult to distinguish a competitive market bid from a bid made through the exercise of market power. As explained by [7], it would be profitable for a company to make offers that far exceeds their marginal cost in hopes that some of their capacity will not be dispatched (the operator will always choose to dispatch the cheapest capacity first). However, in times of scarcity, the prices will also be higher. This could lead to a generator offer being disregarded as economic withholding and being excluded from competing during scarcity conditions. Even though the offer caps would be increased, a regulator

of the future might revert it, which naturally leads to lower incentives for investments.

Furthermore, as mentioned by [8] and [9], high penetration of renewable energy sources (RES) will lead to lower energy prices due to their low production costs. This would make it harder for existing generators to cover their fixed costs, which could further add to the adequacy problem and, in turn aggravate the Missing Money Problem.

B. Missing Money Problem

For an investment in new generation to be economically viable, the clearing price (the price where supply and demand intersects) must be higher than the marginal costs for a certain amount of time. If not, the investment will not be able to recover the capital cost. As mentioned by [10], the introduction of political and administrative restrictions such as price-caps will lead to market distortions that will prevent the prices from rising to a sufficient level for new generation to be procured.

As discussed above, a price-cap is often used to prevent prices from rising above political stable levels. The difference in income from the uncapped market versus the capped market is called "Missing Money", as illustrated in Figure 1.

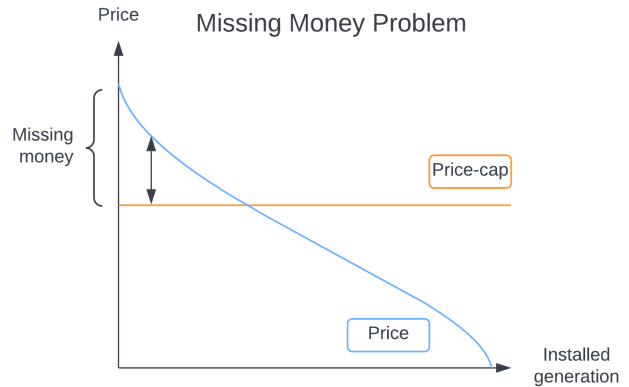


Fig. 1. Illustration of the Missing Money Problem. When demand is higher than the price-cap, suppliers will miss out on potential revenue.

C. Capacity markets

To combat the problem with the adequacy problem, many countries have introduced some kind of capacity market to complement the existing energy-only market. The idea is that producers get paid for their ability to provide energy, even though they may not deliver any. This could serve as a tool to solve the Missing Money Problem since producers could find it profitable to invest in capacity instead.

A capacity market can be set up in different ways. Some markets will have an auction where generators will send offers based on their marginal costs. The last offer that clears the market will set the price for every accepted bid. Bids that exceeds this price will not be payed, while every other capacity offer will get the market clearing price. This kind of market auction is something that is commonly used in the US today.

Instead of an auction, capacity can also be procured through administrative measures such as *Strategic Reserves*. In a strategic reserve, the system operator will contract generators to be kept out of the energy market and be held in reserve. This reserve will then be dispatched in the situation of inadequate generation.

Both capacity auctions and strategic reserves have in common that the capacity in the grid is being planned years ahead in what is called a *forward market*. In addition to forwarding markets, there are ways to procure capacity on the real-time market through what is called *Operational Reserves*.

D. Demand curves

Regardless of which capacity market is being used, there is a need to simulate the demand for capacity. This stems from the fact that customers are not participating in the capacity market; they buy electricity, not capacity. Therefore we need to estimate what customers would be willing to pay for the capacity to get an estimation of the actual demand for grid reliability. To do this, we first need to know the economic costs of a blackout occurring. This cost is called the Value of Lost Load (VOLL) and represents lost revenue from unserved energy, production halts in factories, etc. This value also includes the social cost for households in the form of stress and inconvenience. This value is notoriously difficult to estimate, according to [11], which makes it a rough estimation.

As Adriaan van der Welle and Bob van der Zwaan explain in [12] several different methods exist to estimate VOLL. One of the methods mentioned by the authors is the use of “Stated preferences”; a method in which a questionnaire is sent out and the subject is asked to state how much they are willing to pay to avoid blackouts (or compensation in lower bills to accept blackouts from happening). As demonstrated in [11], we can together with a value of Loss of Load Probability (LOLP), estimate the demand for reliability/capacity.

However, there are other ways to create demand curves. One standard method is to use the value of Cost of New Entry (CONE), which represents the level where new generators are expected just to recover their capital cost and fixed operating costs. PJM is offering offsets to reliable resources such as hydro, coal and gas etc which can provide Energy and Ancillary Services (E&AS) such as acting as frequency regulation. This value is then subtracted from CONE to get a Net-CONE, which represents how much money a new generator will need to earn in the capacity market for investors being able to invest in new generation. This is also the method that is used by PJM. The value for Net-CONE greatly depends on the type of generator in question. Generators with high investment costs, such as nuclear power, will need to earn more on the market compared to low-cost investments such as wind power.

E. PJM and the VRR curve

In PJM, capacity is procured through a three year forward market. In this market, generators will submit offers based on their cost of operation, including capital costs. When an offer is submitted, the generator is committed to having that

production capacity available three years ahead. Bids are then organized from lowest to highest until the demand meets the quantity, at which point it becomes the *clearing price*. Every megawatt lower than this will get paid the clearing price instead of the actual bid as illustrated in Figure 2.

As a way to simulate the demand side, PJM currently uses a demand curve called Variable Resource Reserve (VRR). As mentioned in [13], this curve is designed as a downward slope based on three different points as illustrated in Figure 3. These points are based mainly on values for Net-CONE and a chosen level of reliability that corresponds to a blackout once every ten years. This reliability corresponds with the Installed Reserve Margin (IRM), which according to [14], is defined as “the level of installed reserves needed to maintain a loss of load expectation of one occurrence every ten years”.

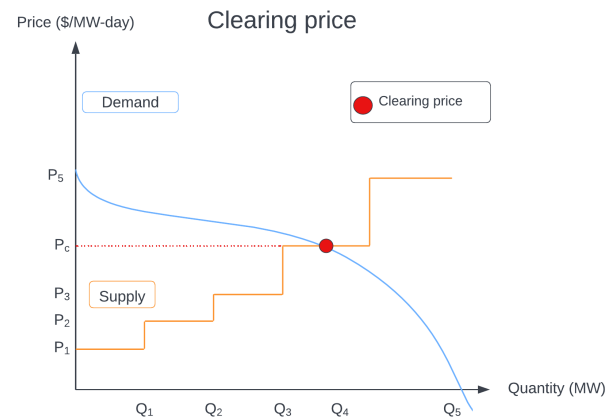


Fig. 2. Quantity Q_1 , Q_2 , Q_3 , Q_4 , and Q_5 are offered at P_1 , P_2 , P_3 , P_4 , and P_5 , respectively. The demand curve intersects the supply curve at Q_4 at the price P_4 , which then becomes the clearing price (P_c). Q_5 was too expensive and will not be sold.

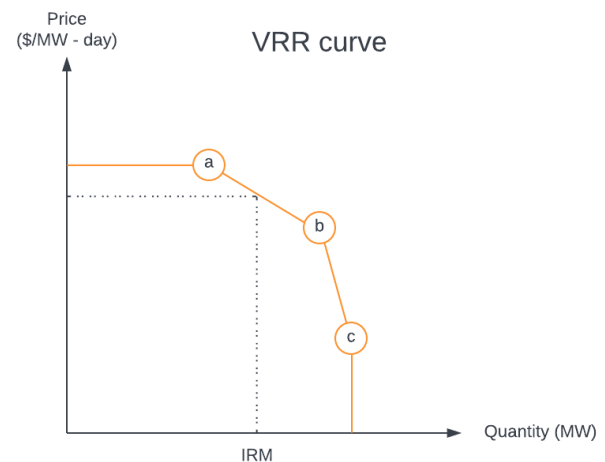


Fig. 3. Illustration of the design of PJM's VRR curve. The three reference points (a), (b), and (c) are marked-out respectively

As we can see from Figure 3, the price for capacity beyond the IRM is above zero; the system still values capacity above what is considered needed. This design differs from what could be considered to be a more intuitive approach with the use of a vertical slope at the IRM instead. The vertical demand curve tells the market that we are only willing to pay for capacity when we are in a deficit. When we procured enough capacity to meet the reliability standards, we no longer want to buy.

However, a vertical demand curve can create problems with volatile prices. As discussed by [15], a subtle change in availability creates massive jumps in prices. This incentivizes the use of market-power through economic withholding. As illustrated in Figure 4, a power plant could refrain from selling some of its capacity with the knowledge that the prices will go up. This holds true even for smaller actors, since only a subtle change in supply will inflate the prices.

In comparison, a downward slope would reduce this risk, since it would require more capacity to be curtailed in order to create a profitable change. As discussed in [16], this is the main idea behind PJM's VRR curve and why it clears capacity greater than IRM. One other key features the author mentions is that it avoids the risk of any shortfalls due to unexpected weather conditions or increased consumption.

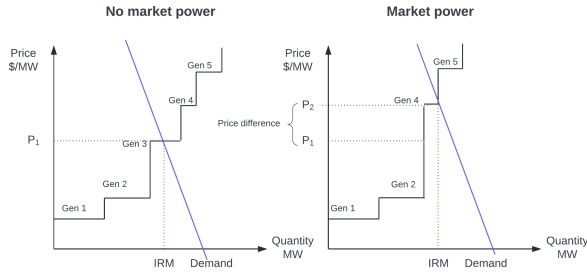


Fig. 4. Illustration of the effects of market power on prices. Generator 3 refrains from offering its capacity in order to inflate the clearing price. A company who owns multiple generators could therefore earn more profit from the other generators.

III. IMPLEMENTATION

The simulation model consists of nine different Locational Deliverability Area zones (LDA) that is part of PJM: DAY, MAAC, SWMAAC, PEPCO, BGE, PPL, DEO&K, EMAAC and ComEd, see Figure 5. The LDA zones are sub-regions within PJM that is used for evaluating locational constraints, such as transmission capacity. Each zone is participating in the auction, and the their capacity bids are shown in Table III.

Nine different resource types are included in the simulation: Nuclear (NUC), Coal (COA), Combined Cycle (CYC), Combustion Turbine (TUR), Solar PV Tracking (TRA), Solar PV Fixed (FIX), Onshore Wind (ONS), Offshore Wind (OFF) and Battery Storage (BAT).

Based on information from the LDAs, PJM is calculating the levels of capacity needed in the system (IRM and RelReq) and an estimation of the probability that a generator will not be available when there is a demand to generate (EFORd), see Table I.

The capacity market for PJM uses VRR (*Variable Resource Reserve demand*) curve and *Supply curve* to solve the Clearing Quantity and Clearing Price for each resources type. The VRR model used in the simulation is retrieved from [17] and the *Supply curve* model is retrieved from [18]. This models is created by PJM based on their own experiences on the market.

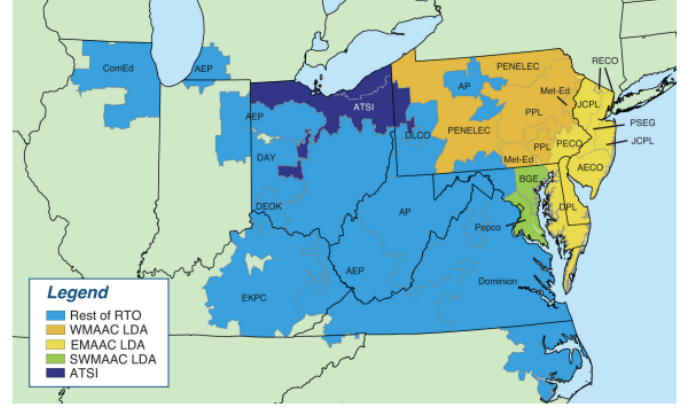


Fig. 5. A map over PJM and its different LDA zones. Image source: PJM Monitoring Analytics [19]

A. Variable names

$E\&AS_i$	E&AS Offset for i (\$/MW-Day)
$CONE_i$	CONE for i (\$/MW-Day)
$EFORd_{RTO}$	Pool-Wide Average EFORd (%) in RTO
IRM_{RTO}	Installed Reserve Margin (%) in RTO
$RelReq_{RTO}$	Reliability Requirement (MW) in RTO
a_{xi}	Unforced Quantity IRM -3% (MW) in i
b_{xi}	Unforced Quantity IRM +1% (MW) in i
c_{xi}	Unforced Quantity IRM +5% (MW) in i
a_{yi}	Price 1.5xNet-Cone (\$/MW-day) in i
b_{yi}	Price 1.0xNet-Cone (\$/MW-day) in i
c_{yi}	Price 0.2xNet-Cone (\$/MW-day) in i
q_n	Unforced Quantity (MW) in n
C_n	Capacity price (\$/MW-day) in n

B. Variable Resource Reserve demand

The model for the VRR curve consists of three different points (a,b and c) based on Unforced Quantity [MW] and prices [\$/MWh]. The Unforced Quantity is being defined by [20] as "...the amount of a generator's total capacity that is allowed to count as firm capacity in the auction. It represents the percentage of weather-adjusted installed capacity available after a unit's expected outage rate is taken into account".

The Unforced Quantity will be describe on the x -axis and the price will be described on the y -axis. Lines connecting from zero-a, a-b, b-c and c to x -axis is then created.

X-axis:

$$\begin{aligned} a_{xi} &= RelReq_{RTO} \times \left(\frac{1 + IRM_{RTO} - 3\%}{1 + IRM_{RTO}} \right) \\ b_{xi} &= RelReq_{RTO} \times \left(\frac{1 + IRM_{RTO} + 1\%}{1 + IRM_{RTO}} \right) \\ c_{xi} &= RelReq_{RTO} \times \left(\frac{1 + IRM_{RTO} + 5\%}{1 + IRM_{RTO}} \right) \end{aligned} \quad (1)$$

Y-axis:

$$\begin{aligned} a_{yi} &= 1.5 \times \left(\frac{CONE_i - E\&AS_i}{1 - EFORD_{RTO}} \right) \\ b_{yi} &= 1.0 \times \left(\frac{CONE_i - E\&AS_i}{1 - EFORD_{RTO}} \right) \\ c_{yi} &= 0.2 \times \left(\frac{CONE_i - E\&AS_i}{1 - EFORD_{RTO}} \right) \end{aligned} \quad (2)$$

Basis points:

$$a_i = (a_{xi}, a_{yi}) \quad b_i = (b_{xi}, b_{yi}) \quad c_i = (c_{xi}, c_{yi}) \quad (3)$$

C. Supply Curves

The bids from the capacity auctions can be represented as a step function. This is based on Unforced Quantity requirement and Capacity Price from LDA and adding as (q_n, C_n) in the system, while q_n is the quantity required and C_n is the price. The function starts with the lowest price and ends with the highest. The (x_n, y_n) points gives the location in the graph where the LDA is placed in the step function in the chart and vertical and horizontal lines are then connecting these points. The description for x_n and y_n are following:

$$\begin{aligned} x_n &= x_{n-1} + q_n \quad ; x_0 = 0 \\ y_n &= C_n \end{aligned} \quad (4)$$

There are two supply curves methods that can be used together with the VRR demand curve: *The Smoothing Method* and *The Polynomial Form*. Both of these methods are retrieved from [18]. The first one, *The Smoothing Method*, gives a smooth approximation between the step function that is based on an exponential formula function with A and B being coefficients.

$$C(Q) = A \times e^{B \times Q} \quad (5)$$

The second one, *The Polynomial Form*, is a polynomial equation function. The problem with this method is to decide how many polynomial terms needed in order to get the best approximation. In the equation below, a_{n-1} is a coefficient.

$$C(Q) = a_0 + a_1 \times Q + a_2 \times Q^2 + a_3 \times Q^3 \dots \quad (6)$$

D. MATLAB implementation

The data in Table I and Table III is retrieved from the Base Auction Residual Results from 2020/2021 [21] and the data for Table II is retrieved from [22]. These values was then used to create VRR curves for different resource types.

TABLE I
PARAMETERS FOR VRR IN RTO

Reserve Requirement	2020/2021 BRA
IRM_{RTO}	16.6%
$EFORD_{RTO}$	6.59%
$RelReq_{RTO}$	156239.5 MW

TABLE II
PARAMETERS FOR VRR

Resource	$CONE_i$	$E\&AS_i$
NUC	2000	517
COA	1068	43
CYC	320	168
TUR	294	48
TRA	290	185
FIX	271	117
ONS	420	240
OFF	1155	337
BAT	532	116

TABLE III
PARAMETERS FOR STEP FUNCTION

n	LDA	q_n (MW)	C_n (\$/MW - Day)
1	DAY	3850.6	76.53
2	MAAC	65138.7	76.53
3	SWMAAC	14964.3	86.05
4	PEPCO	7314.4	86.05
5	BGE	7649.9	86.05
6	PPL	8238.3	86.05
7	DEO&K	5205.5	130.00
8	EMAAC	35369.6	187.88
9	ComEd	25153.0	188.13

To solve *The Smoothing Method* coefficients A and B, the MATLAB-function *fit* was used. This function approximates the coefficients based on the input values from the step function. To use this function, two vectors x_i and y_i is created. These vectors contains every X variables and Y variables respectively. The vectors was then transposed to be able to work with the function *fit*. The calculated coefficients are shown in Table V.

$$X = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8 \ x_9]^T \quad (7)$$

$$Y = [y_1 \ y_2 \ y_3 \ y_4 \ y_5 \ y_6 \ y_7 \ y_8 \ y_9]^T$$

To solve the a_j coefficients for *The Polynomial Form*, the MATLAB-function *polyfit* was used. However, the *polyfit* function do not require the vectors to be transposed as in the case with *The Smoothing Method*.

$$X = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8 \ x_9] \quad (8)$$

$$Y = [y_1 \ y_2 \ y_3 \ y_4 \ y_5 \ y_6 \ y_7 \ y_8 \ y_9]$$

IV. SIMULATION RESULTS

A. Supply Curves

The results from the step function described in section III-C is presented in Table IV and the calculated coefficients for both the supply curves, are presented in Table V.

Figure 6 shows the results of all three curves. For *The Polynomial Form*, a third degree polynomial was chosen, since higher order polynomials gave a more sinus looking curve. However, *The Smoothing Method* gave a better result than *The Polynomial Method*. This is because in an auction, the prices goes up, and with the polynomial function the curve was sloping downward at the start. Therefore, the *Smoothing method* was chosen for the simulations.

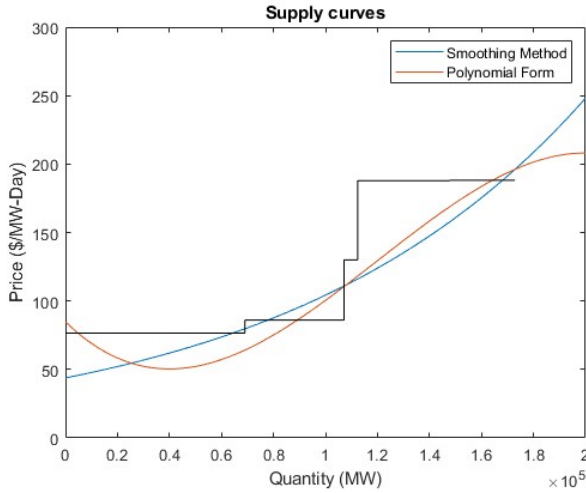


Fig. 6. The figure shows the step function combined with the approximated supply curves. As can be seen in the graph, the smoothing method is more adaptable for VRR demand as it approximate closer to LDAs step function

TABLE IV
RESULTS FOR STEP FUNCTION

n	LDA	$x_n (MW)$	$y_n (\$/MW - Day)$
1	DAY	3850.6	76.53
2	MAAC	68989.3	76.53
3	SWMAAC	83953.6	86.05
4	PEPCO	91268.0	86.05
5	BGE	98917.9	86.05
6	PLL	107156.2	86.05
7	DEO&K	112361.7	130.00
8	EMAAC	147731.3	187.88
9	ComEd	172884.3	188.13

TABLE V
COEFFICIENT VALUES FOR BOTH SUPPLY CURVES

Coefficients	Value
A	43.82
B	8.668e-06
a_0	84.8304
a_1	-0.0018
a_2	2.7756e-08
a_3	-7.7182e-14

B. Clearing Quantity and Price

Figure 7 and 8 shows the result from combining the VRR curves together with the *Smoothing Method* as supply curve.

Table VI-VIII shows the values for the calculation of the points *a*, *b* and *c* and Table IX shows the results of clearing quantities and clearing prices for each resource when the curves intersect with each other.

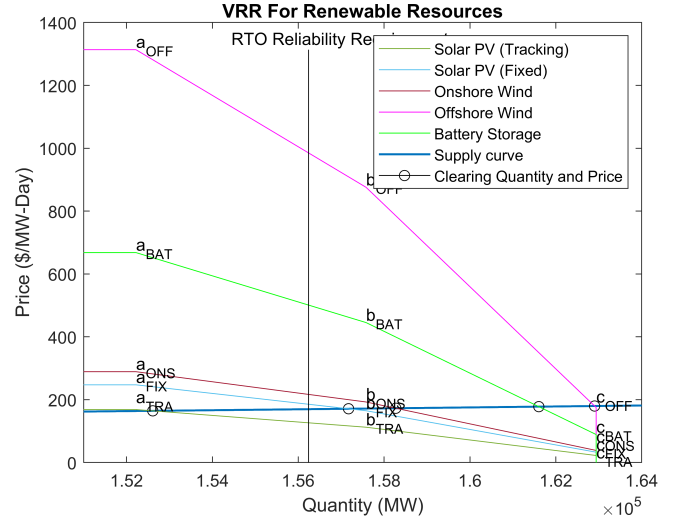


Fig. 7. The demand curve for renewable resources is adapted from section III-B and section III-C with parameters from Table II

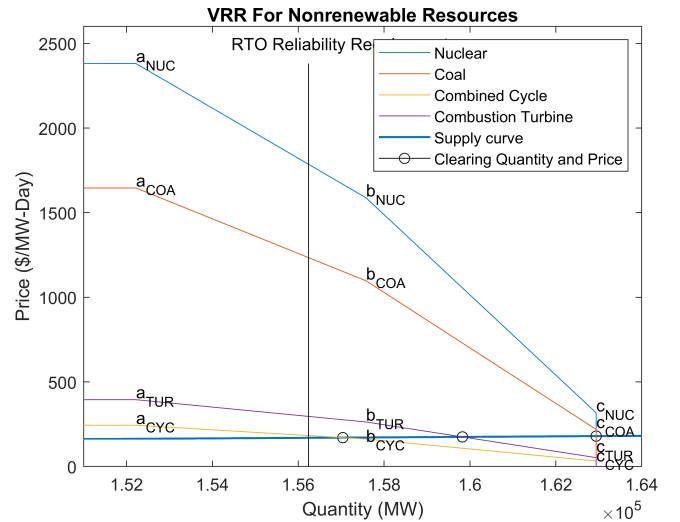


Fig. 8. The demand curve for nonrenewable resources is adapted from section III-B and section III-C with parameters from Table II

TABLE VI
RESULTS FOR POINT a_i IN VRR

a_i	$a_{x,i}$	$a_{y,i}$
NUC	152220	2381.40
COA	152220	1646.00
CYC	152220	244.10
TUR	152220	395.00
TRA	152220	168.60
FIX	152220	247.30
ONS	152220	289.00
OFF	152220	1313.60
BAT	152220	668.00

TABLE VII
RESULTS FOR POINT b_i IN VRR

b_i	b_{xi}	b_{yi}
NUC	157580	1587.6
COA	157580	1097.3
CYC	157580	162.7
TUR	157580	263.4
TRA	157580	112.4
FIX	157580	164.9
ONS	157580	192.7
OFF	157580	875.7
BAT	157580	445.3

TABLE VIII
RESULTS FOR POINT c_i IN VRR

c_i	c_{xi}	c_{yi}
NUC	162940	317.53
COA	162940	219.46
CYC	162940	32.54
TUR	162940	52.67
TRA	162940	22.48
FIX	162940	32.97
ONS	162940	38.54
OFF	162940	175.14
BAT	162940	89.07

TABLE IX
RESULTS FOR CLEARING QUANTITY AND PRICE

Resource	Quantity (MW)	Price (\$/MW-Day)
NUC	162940	179.91
COA	162940	179.91
CYC	157040	170.94
TUR	159820	175.12
TRA	152610	160.50
FIX	157170	171.14
ONS	158270	170.78
OFF	162900	179.85
BAT	161600	177.84

V. DISCUSSION

A. Clearing Price and CONE

From the results we can see that both nuclear and coal got a clearing price of 179.91 \$/MW-Day. This was highest price out of the simulated resources. In comparison, the lowest clearing price was cleared by solar tracking at 160.50 \$/MW-Day. The difference could be explained by the fact that the cost of entering the market is much higher for nuclear power. This is due to the large capital investments needed for construction. Furthermore, it takes long time to build and have a high maintenance cost. Moreover, the low E&AS offsets for coal elevates the demand curve for coal closer to that of nuclear, which can explain the high clearing price.

B. Effects on energy prices due to Covid-19

When the US declared emergency in March 2020, the people was sent to lockdown. This made the power demand to drop and lower the average marginal price which peaked at \$17.50/MWh in April 2020. This is a downfall about 1/3 from the prices from April 2019 according to [23]. The natural gas was extremely cheap and Texas had the biggest price decline for gas with 36.5% lower prices compared to the price levels of year 2019. Gas generation was not the only generation that decreased, even Coal generation fell behind.

In contrast, the production of nuclear power increased by 2-3%. One possible explanation is that the US does not have a large scale renewable generation compared to other countries. In a report from [24], PJM's coal generation decreased with 40% compared to 2019. On the other hand, solar generation instead increased 45% during the same time.

C. War between Ukraine and Russia

At the beginning of the year 2022 Russia decided to invade Ukraine. This caused a significant impact on the natural gas import and export, especially in Europe. Russia is one of the biggest natural gas exporters, and many other European countries have limited access to natural gas. Natural gas plays a significant role in the electricity production, especially in the US. According to [25], natural gas made up 38.3% of the US energy production in year 2021.

As described by [26], the president of the US, Joe Biden, and the congress passed a ban of importing gas, oil and coal from Russia to stop Russia's invasion of Ukraine. This may lead to increased cost in the production for Combustion Turbine (TUR) and Combined Cycle (CYC) that uses natural gas as fuel.

As of now, EU is highly dependent on importing natural gas from Russia. According to [27], Russia supplies EU with around 40% of EU's natural gas needs. In an attempt to make EU less reliant on Russia, the US has sign an agreement to provide EU with around 10% of the gas that they currently import from Russia according to [27]. This may cause electricity production in the US to be shorted of natural gas and in turn increasing the prices. According to [28], the prices for natural gas in US almost doubled from \$13.65/MWh at the beginning of the year 2022 to \$24.81/MWh by the end of April same year.

Furthermore, the production of renewable sources for the year 2021 was estimated at 20.1%, which is too low to compensate for the exported natural gas. Instead, we might see that nuclear production will increase to cover the energy demand, especially during winter time when the energy demand is higher.

D. Social-welfare

We could see from the results that the use of a forward capacity market is successful at procuring capacity. However, when we predefine a curve such as the one from PJM, the question about what is a desired level of capacity is already included in its construction.

A question that arises: is forward capacity markets *effective* at procuring capacity? In other words, are they effectively maximizing the social-welfare so that we pay the least amount of money for the greatest level of capacity?

Some are criticizing these markets of overpaying as well as procuring greater capacity levels than needed. For every megawatt procured that we do not use, we have essentially wasted money and thus lowered the social-welfare. On the other hand, blackouts and involuntary curtailment are not only expensive in monetary terms, it is also expensive from a social point of view.

Others have criticised capacity markets for just acting as a band-aid for the implementation of offer caps. Without offer caps, even the most expensive units could cover their capital costs. One could argue, as has been done before, that a more optimal solution would be to adjust the offer caps upwards to allow more capacity to stay in the market. This is what has been done in the Texas market, ERCOT.

This paper has only studied the concept of forward capacity markets. As mentioned earlier, there exists a variety of market concepts which seek to aid the procurement of capacity.

VI. CONCLUSIONS

While the general consensus about how to stop the global warming is to include more and more renewable energy in the system, there seems to be no easy solution in how to implement it. The more renewable there are in the system, the harder it becomes for thermal generation such as nuclear, coal and gas to stay in the market due to renewable generation having near zero marginal costs. Furthermore, we are still not technically prepared for a energy system with 100% renewable sources, due to reliability issues. However, as proposed earlier, a capacity market could be the solution to the lack of capacity in the system. As we can see from the results, such a market could indeed, in theory, be successfully at procuring enough capacity for the grid.

APPENDIX A

MATLAB script for the Supply Curves and Variable Resource Reserve demand curves

ACKNOWLEDGMENT

We would want to share our gratitude to our supervisor Mohammad Reza Hesamzadeh (Division of Electric Power and Energy Systems, KTH) for his guidance and feedback for this project.

REFERENCES

- [1] S. Lazarou, C. Christodoulou, and V. Vita, "Global change assessment model (gcam) considerations of the primary sources energy mix for an energetic scenario that could meet Paris agreement," in *2019 54th International Universities Power Engineering Conference (UPEC)*, 2019, pp. 1–5.
- [2] —, *Efficient Investment in Generation and Consumption Assets*. John Wiley Sons, Ltd, 2014, ch. 9, pp. 181–198. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118775745.ch09>
- [3] S. Lazarou, C. Christodoulou, and V. Vita, "Global change assessment model (gcam) considerations of the primary sources energy mix for an energetic scenario that could meet Paris agreement," in *2019 54th International Universities Power Engineering Conference (UPEC)*, 2019, pp. 1–5.
- [4] P. Cramton, A. Ockenfels, and S. Stoft, "Capacity market fundamentals," *Economics of Energy Environmental Policy*, vol. Volume 2, no. Number 2, 2013. [Online]. Available: https://EconPapers.repec.org/RePEc:aen:eeepjl:2_2_a02
- [5] J. Wu, X. Guan, F. Gao, and G. Sun, "Social welfare maximization auction for electricity markets with elastic demand," in *2008 7th World Congress on Intelligent Control and Automation*, 2008, pp. 7157–7162.
- [6] R. Hase and N. Shinomiya, "Maximization of social welfare in deregulated electricity markets with intermediaries," in *2015 11th International Conference on Innovations in Information Technology (IIT)*, 2015, pp. 256–261.
- [7] F. Zhang and H. Zhou, "Research on economic withholding in wholesale markets based on incremental heat rate," in *2005 IEEE/PES Transmission Distribution Conference Exposition: Asia and Pacific*, 2005, pp. 1–7.
- [8] G. Gallo, "Electricity market manipulation: How behavioral modeling can help market design," Dec. 2015. [Online]. Available: <https://www.nrel.gov/docs/fy16osti/65416.pdf>
- [9] P. C. See, O. B. Fosso, K. Y. Wong, and M. Molinas, "Flow-based forward capacity mechanism: An alternative to the regulated capacity remuneration mechanisms in electricity market with high res penetration," *IEEE Transactions on Sustainable Energy*, vol. 7, no. 2, pp. 830–840, 2016.
- [10] L. John Wiley Sons, *Market-Based Investment in Electricity Generation*. John Wiley Sons, Ltd, 2014, ch. 10, pp. 199–208. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118775745.ch10>
- [11] 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.
- [12] A. van der Welle and B. van der Zwaan, "An Overview of Selected Studies on the Value of Lost Load (VOLL)," Tech. Rep., Nov. 1942. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.492.792&rep=rep1&type=pdf>
- [13] The Brattle Group. (2014, May) Third triennial review of pjm's variable resource requirement curve. [Online]. Available: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.641.2055&rep=rep1&type=pdf>
- [14] S. Benedettini. (2013, Jul.) Pjm and iso-ne forward capacity markets: a critical assessment. [Online]. Available: <https://www.pjm.com/~media/documents/manuals/m20.ashx>
- [15] F. A. Felder and C. J. Loxley. (2022, Mar.) The implications of a vertical demand curve in solar renewable portfolio standards. [Online]. Available: <http://cecep.rutgers.edu/wp-content/uploads/2013/11/VerticalDemandCurve.pdf>
- [16] S. Benedettini. (2013, Jul.) Pjm and iso-ne forward capacity markets: a critical assessment. [Online]. Available: <https://green.unibocconi.eu/sites/default/files/media/attach/Report-EC.pdf>
- [17] W. W. Hogan. (2017, Sep.) Electricity market design interactions of multiple markets. [Online]. Available: https://media.rff.org/documents/170914_PowerMarkets_WilliamHogan.pdf
- [18] K. Dorko and J. Bowring. (2013, Jul.) Smoothing the rpm supply curve. Monitoring Analytics Study. [Online]. Available: <https://www.pjm.com/~media/committees-groups/committees/mic/20131106-fmu/20131106-item-03-imm-supply-curve-smoothing.ashx>
- [19] Monitorin Analytics. (2013, Mar.) State of the market report for pjm. [Online]. Available: https://www.monitoringanalytics.com/reports/PJM_State_of_the_Market/2014/2014-som-pjm-volume2-sec5.pdf
- [20] M. Rose. (2022, May) Capacity market auction results and rule changes in pjm. "Enerdynamics Corp, Colorado, CO, USA". [Online]. Available: https://www.enerdynamics.com/Energy-Currents_Blog/Capacity-Market-Auction-Results-and-Rule-Changes-in-PJM.aspx
- [21] PJM. (2021, Aug.) 2020/2021 base residual auction results. Results from 2020-2021 auction. [Online]. Available: <http://www.pjm.com/~media/markets-ops/rpm/rpm-auction-info/2020-2021-base-residual-auction-results.ashx>
- [22] Market Implementation Committee. (2020, Mar.) Default mopr floor offer prices for new generation capacity resources. PJM, US. Resources Capacity Study. [Online]. Available: <https://www.pjm.com/~media/committees-groups/committees/mic/2020/20200311/20200311-item-06c-default-mopr-cone.ashx>
- [23] M. Watson. (2020, May) Pjm tracker: Pandemic-weakened power demand, cheap gas sap power prices. S&P Global, New York, NY. [Online]. Available: <https://www.spglobal.com/commodityinsights/en/market-insights/latest-news/electric-power/050720-pjm-tracker-pandemic-weakened-power-demand-cheap-gas-sap-power-prices>
- [24] B. Alves. (2021, Jul.) Year-on-year change in pjm's power generation due to coronavirus (covid-19) lockdowns in the united states in 2020, by energy source*. Statista, New York, NY. [Online]. Available: <https://www.statista.com/statistics/1117567/pjm-us-power-generation-year-on-year-change-covid-19-lockdowns-by-type/#statisticContainer>
- [25] U.S. Energy Information Administration. (2022, Mar.) What is u.s. electricity generation by energy source? U.S. Energy Information Administration, Washington, DC. [Online]. Available: <https://www.eia.gov/tools/faqs/faq.php?id=427&t=3>
- [26] C. Wilkie. (2022, Apr.) Congress passes ban on russian oil and gas imports, sending measure to Biden. CNBC, New Jersey, NJ. [Online]. Available: <https://www.cnbc.com/2022/04/07/senate-passes-ban-on-russian-oil-and-gas-imports-.html>

- [27] BBC. (2022, Mar.) Eu signs us gas deal to curb reliance on russia. BBC, United Kingdom, UK. [Online]. Available: <https://www.bbc.com/news/business-60871601>
- [28] M. Grossman. (2022, Apr.) Russian supply concerns drive natural-gas prices higher. "Wall Street Journal, New York, NY". [Online]. Available: <https://www.wsj.com/articles/russian-supply-concerns-drive-natural-gas-prices-higher-11651077617>

Capacity Market Design and Theory

Simon Palmborg and Lisa Thor

Abstract—Most modern electricity markets do not guarantee that generation is reliable and sufficient to provide all consumer's electricity needs at all times. This is due to design-flaws and regulatory intervention. During the coming decades, increased electricity demand and decarbonization trends will affect the electricity market greatly. As the share of wind and solar power increases in the generation mix, the inertia in the power system is expected to decrease. This can potentially increase the systems exposure to blackout risk. Therefore, it is important to ensure that electricity is traded in a way that ensures enough supply even during scarcity events. The study aims to compare six different capacity market designs that are widely discussed in the scientific literature. Furthermore this study uses MATLAB to simulate how the utility for the strategic reserve in Sweden has changed over the past few years. The study finds no ideal capacity market design, but concludes that different solutions come with their own advantages and trade-offs. The simulation results show that the utility of the strategic reserve in Sweden has increased during the last few years. Additionally the simulation results suggest that demand for the strategic result varies on a daily time frame.

Sammanfattning—De flesta moderna elmarknader kan inte garantera pålitlig och tillräcklig produktion för att tillgodose konsumenters elbehov under alla tillfällen. Detta har att göra med brister i designen av elnätet och regleringar. Under de kommande decennierna kommer trender inom avkarbonisering att ha stor inverkan på elmarknaden. Med en ökande andel vind- och solkraft som kraftproduktionslag förväntas trögheten i kraftsystemet att minska. Detta kan potentiellt höja systemets utsatthet för strömavbrott. Därför är det viktigt att el handlas på ett sätt som försäkrar tillräcklig elförsörjning även under fall då produktionen är begränsad. Med målet att jämföra sex olika designar av kapacitetsmarknader som är etablerade i tidigare forskningsstudier. Vidare använder denna studie MATLAB för att simulera hur behovet av effektreserven i Sverige har ändrats under de senaste åren. Studien finner ingen ideal design för en kapacitetsmarknad, men fastställer att olika lösningar har sina egna fördelar och avvägningar. Studien finner vidare att behovet av effektreserven har ökat under de senaste åren. Dessutom indikerar att behovet av effektreserven varierar med elbehovet under dagen.

Index Terms—electricity markets, capacity markets, capacity market design, the adequacy problem, strategic reserve.

Supervisor: Mohammad Reza Hesamzadeh

TRITA-EECS-EX-2022:146

I. INTRODUCTION

It is hard to imagine civilization as we know it today without electricity. We need it to heat our buildings, refrigerate our food, light our homes, streets and cities, charge our phones and laptops and power public trains and subways. With a growing world population and electrification trends motivated by global warming concerns, it is a thrilling moment in history to study electricity networks. As the demand for sustainably produced

electricity is expected to increase dramatically over the coming decades it is desired to have an efficient market design for the buying and selling of electricity.

In the late 19th century, electricity legitimately became a competitive energy alternative to steam power and coal. This was due to technological improvements of the coil which led to the invention of the dynamo, and large-scale electricity generation became commercially feasible. The already well-established markets of coal and steam power troubled electricity to find a market of its own, but with the invention of the light bulb 1877, a commercial use case was found [1]. Infrastructure necessary for transmitting electricity throughout society was driven by rapid industrialization and an innovative scientific revolution in electrical science and social life. The AC-concept discovered by Nikola Tesla in the late 19th century simplified voltage-conversion by using a transformer. This breakthrough led to large scale adoption followed by implementation of centralized large industrial-scale power plants [2].

While the world has seen massive technological developments in IT- and communication services, power plant efficiency and electricity distribution hasn't changed much [1] [3]. The modern power grid and distribution system is based entirely on the fundamental principles coined by Tesla in the late 19th century [1]. Until only a few decades ago, the primary generation of electricity was made in large industrial plants and distributed one-way to consumers via transmission and distribution networks [1] [3]. Today, the modern electricity industry is undergoing a fundamental shift in structure [3].

Decarbonization trends in the energy sector along with national political interest of becoming energy independent is pressuring implementation of renewable energy production. Societal shifts and technological development are furthermore impacting the transition in the energy sector. Advancements in battery technology are creating possibilities in the way electricity is stored and consumed. In addition, the car industry is electrifying, and electric vehicles are creating additional load in the distribution system. Developments in the IT and communication sector are changing the way consumers act in the electricity market, with innovation in appliances and devices allowing consumers to participate more actively in the market. This is imagined to threaten the traditionally centralized structure of the energy system as customers integrate to actively respond to local market conditions and consumers have the option to produce and sell their locally generated electricity [3].

As a means to achieve efficient economic outcomes in electricity trading, modern society has implemented a competitive market structure. This has the intention to optimize the economic outcome as electricity is sold and purchased. The desired outcome is for producers to achieve high marginal profit, while consumers derive high marginal utility from

the electricity bought. Through optimization an ideal market outcome can be modeled that maximizes the sum of marginal utility for consumers and marginal profit for producers. Although this optimization surely can be applied to electricity markets, electricity networks are complex in their nature and require the central role of a system operator to function. In the same exact moment electricity is used on the consumer end of an electricity network, a power plant must instantly produce the same amount of electricity. This criterion also includes transportation of energy that is available between the system nodes. It is thus not possible to separate the market for the transportation of electric power and the market for production or consumption of power [3].

The need for capacity markets stems from various different market failures in the electricity market, one of them being the consumers' limited ability to react to electricity price changes. The electricity market is inelastic in both the demand- and supply side of the market. The consumers are dependent on electricity and have a limited ability to both reduce and plan their consumption. The supply side can not store electricity because of high storage costs, it is hard for the supply side to immediately meet demand. This leads to an inability of the market to determine the efficient level of generation capacity, which in turn leads to blackouts in times of scarcity. As the electricity market changes in both structure and size, it is important to counteract this to have reliability in the grid. Investments in capacity means investments in electricity with reliability [4].

II. MOTIVATION

A. The problem with renewable energy

As Sweden and the rest of the world transition to renewable electricity production, we expect to see an increased amount of wind and solar generation that has experienced rapid innovation and improvement during the last few decades. With a decarbonization of the energy sector, we enter a new paradigm of intermittent energy output replacing otherwise reliable energy production in the forms of coal, gas, oil and in Sweden's case: nuclear power. Due to subsidies and low incremental costs, the supply of renewables are not price-sensitive. This, in turn, means that renewables contribute to the problem of price-inelastic demand, which is a fundamental reason for implementing capacity markets [4] [3].

Wind- and solar power, without implementation of batteries, are unable to provide reliable electricity generation. Therefore, development in renewable power generation can only partly serve as a replacement for conventional sources such as coal and gas. Although, with additional implementation of batteries, intermittent renewable generation would make the power generation reliable. Furthermore, the price volatility induced by intermittent generation tends to market price levels which is disadvantageous for conventional capacity, adding to the adequacy problem [4].

The repercussions of the stated above complexities with implementation of renewables result in investors facing uncertainty regarding the future mix of generation, energy prices and energy regulation [4]. In this aspect, an issue can be

seen with how electricity sometimes is priced when generation becomes scarce. As the demand for electricity consumption outpaces that of generation, the price for electricity rises until it reaches a regulatory set price cap. Thus, in a wholesale electricity market, a price cap diminishes the net revenues of the electricity producers [3]. In a way this disincentivizes generation when electricity becomes scarce. Consequently, this lowers the incentives for producing additional capacity. Consequently, the lack of investment incentive is formulated as the 'missing money' problem and results in an inefficient mix of generation. In a capacity market, in order to compensate for the lack of investments in capacity, generators are given additional payments based on available capacity [4].

B. Pandemic aftermath

In the process of declining COVID-19 cases, and with a partly vaccinated population, social restrictions have now eased. Therefore, the economy is now opening back up, increasing the demand for transportation, businesses, restaurants and travel. As society returns to its normal state, so does the energy demand which has seen a steady increase since March and April 2020. While some parts of the world have come further in the reopening of the economy, there is still a lot of room to go before everything is back to normal standards. It is therefore expected that demand for electricity can grow further and in doing so put pressure on generators [5].

C. Nordic energy- and capacity market

The Nordic power market is composed of the individual markets of Sweden, Finland, Denmark and Norway. The trading is formed of three different markets, two of which are a day-ahead and intraday market on Nord Pool, as well as a real-time balancing market operated by the Nordic system operators. The Nordic market is intended to exclude capacity payments, but because of provision of primary control capacity, it is not purely an electricity market. The goal is for the market to be mature enough to provide a sufficient level of supply reliability by itself. This means that if the market is provided with additional levels of power generation, regardless of it being dispatchable or not, the reliability of supply will increase or at least remain the same [6]. However, if the dispatchable power is added as a substitute, the reliability of supply will decrease [7].

D. The adequacy problem

The adequacy problem is the lack of adequate generation in modern electricity markets. To solve the adequacy problem is to ensure enough system capacity in order to minimize blackouts in an economically efficient way [4]. Essentially, there is on one hand an aim for enough capacity in the system to make blackouts a rare event, generally once every 10 years. On the other hand, this should be accomplished with economic efficiency, meaning preventing excess payments to and over-investments in capacity. This definition implies that a market cannot meet all requirements for perfect competition [4]. Another frequently used definition of the adequacy problem is the reliability of supply, the problem of whether or not the producers can supply in due time [8].

III. OBJECTIVES

The aim of the project is to develop a good understanding of the theory of capacity markets through already written literature and simulations. This theory is then implemented on a simple electricity grid example and the capacity market design in Sweden is studied to a further level. Most of the work will be literature studies, but simulations using MATLAB will also be used to show how the capacity market works in Sweden. Capacity markets are important to understand because the electricity grid has some capacity challenges that will grow with time, due to further electrification of major industries, cars and so on.

This study aims to provide a new perspective on capacity market design through comparing different market design choices. More specifically are these adequacy problem-solutions applied on an example with the intention to provide the reader of this paper with a more concrete way to understand the purpose of different capacity market designs. Additionally, the different design choices are evaluated through pros and cons and finally summarized in a table. Lastly, simulations are done to show variation in power production by Sweden's strategic reserve Karlshamnsverket.

IV. DIFFERENT SOLUTIONS TO THE ADEQUACY PROBLEM

A. The application example

To properly compare solutions to the adequacy problem, the solutions will be applied to an example of an electrical system. The chosen electrical system is simple and only consists of a system operator, a power plant and consumers in the form of a factory and some residential buildings. Every solution to the adequacy problem that will be covered in this report, will be applied to this example. The example assumes a warm location with high electricity demand during the summer. Capacity markets designs' general purpose is to increase grid reliability. The factory in the example and the residential consumers will get affected in different ways by a blackout. Generally, factories are very dependent on continuous production, so to interrupt the production because of a blackout in the system during a few hours or even minutes often result in major costly problems for the factories. Residential consumers are often more resistant to intermittent electricity interruptions and are therefore generally not as affected by a blackout.

B. Regulated price cap

1) *Theory:* A price cap on electricity prices limits the rate which a supplier can charge for default tariffs. The value of the price cap can be determined by assuming there is a constant price for electricity. In low demand hours, the real market price is below this artificial fixed price, whereas in high demand hours the real market price is above the fixed price. Eventually, restricting the supply is more economically viable than supplying an extra unit. Meaning that, at a certain price, the cost of producing an extra unit of capacity exceeds the loss of not supplying an extra unit. The economic losses from restricting the supply is called the value of lost load (VoLL) [3]. The VoLL will thereby correspond to the value

that the customers are willing to pay to avoid blackouts [4]. According to [3], the value of the price cap should be VoLL plus the fixed price, which in practice means the VoLL since it is much larger than the fixed price. Most modern electricity markets have an implemented price cap. The difference is the value of the price cap, some systems have chosen a much higher price cap than others [3].

2) *Example:* Let us assume a relatively low price cap is set on the example grid system. In high demand times, when both the factory and residential houses consume a lot of electricity, the price cap will eventually force the power plant to stop further supply. This means that past a certain degree of production, the output of the plant will stagnate due to the price cap making it economically non-viable. This means that the demand and supply isn't equal, eventually resulting in a blackout. In lower demand times the regulated price cap will lower the pricing on the electricity [9]. The factory in the example grid can produce its goods to a lower price, but will not have as steady production rate as if there was no price cap. The residents will get affected in the same way.

3) *Pros:* An advantage with this design is that it has the ability to in a relatively easy way adjust the level of the price cap. This can be done in order to fulfill different purposes like more adequate generation, or lower electricity prices. A low price cap reduces risks in the market, which is especially an advantage in countries missing a well developed financial trading market [9].

4) *Cons:* With a regulated price cap, there is always a risk that the price cap is set too low. If the price cap is set too low, blackouts will occur unnecessarily often. A price cap that is set too high can become difficult to handle politically [9]. Determining the value of the price cap can therefore be a difficult task.

Some argue that the adequacy problem can not be solved if there is a regulated price cap, as this fundamentally flaws the market. They mean that to solve the adequacy problem, electricity prices need to be able to get as high as the producers claim during high demand [3].

C. Operating reserve

1) *Theory:* The operating reserve market uses a low price cap and relies on sidelined dispatchable generators to supply capacity when electricity supply becomes scarce. These reserves are generally obligated to respond to changed market conditions, for example mismatched supply and demand [10]. The contract agreements can include certain requirements regarding maximum capacity supply, minimum capacity supply and price of electricity. When the need for sidelined generation arises, an auction will be held with different power plants bidding to be an operating reserve. This allows the market to, though bidding, find an agreement on capacity price. The agreement will also include at what notice the power plant will be obligated to provide capacity. A typical response time is usually granted between 10 and 30 minutes, although this varies between different design choices [11]. Operating reserve markets does not directly incentivize investments in capacity. Instead, it addresses the adequacy problem in a second hand

manner by extrapolating capacity from the wholesale market and thus affecting the wholesale price. Since implementation of an operating reserve would offload demand in the wholesale market, this would increase the wholesale price. Furthermore this would lead to increased profits for regular generators and in turn incentivize new investments [4].

2) *Example:* The power plant in the example would be an operating reserve. The system operator has a contract with the power plant regarding the amount of capacity supplied at what time. Let us say that 90% of the power plants capacity is normally enough for the society. On a very warm day, when all AC:s in the city are at full blast, 100% of the plant's capacity is needed. An operating reserve market mitigates risks that producers would hold back capacity to benefit from scarcity pricing. As an operating reserve, the system operator can make the power plant, within half an hour, produce the amount (100% of the capacity) actually needed. This results in lower and less frequent peak electricity prices and more reliability for the consumers.

3) *Pros:* Operating reserves can be advantageous if implemented with the proper required capacity estimation. This approach is argued to be efficient in solving the adequacy problem, if the demand of operating reserve capacity is expected to be inelastic. Although, the research on this is limited. One method that can be used to optimize the estimation being made is through cost-benefit analysis [10].

4) *Cons:* Although some research, as noted above, suggests more effective implementations of operating reserves, it is more common for these markets to be designed with predetermined capacity. This creates a design that is fundamentally unable to react to demand shift - meaning demand of operating reserve capacity becomes very static [10]. Furthermore, since incentives for adequate generation are not directly addressed through operating reserves, it is argued that there are more efficient ways to incentivize new entry [4].

D. Energy storage

1) *Theory:* Batteries are excellent for storing electrical energy. This technology can be used to even out the supply curve by storing energy during low demand hours and releasing it during high demand hours. Battery systems have different applications, from small home systems to industrial systems and large scale storage systems [12]. [13] Describes how a battery energy storage system (BESS) can benefit the uses for solar panels. A BESS can charge from solar panels at daytime and then discharge the energy in the afternoon, a time where there typically is a high demand. Solar panels can thereby sell the produced energy during high demand hours, when the electricity prices are higher. There is an important distinction to make between the economical value for an amount of MWh to be produced some time, and the same amount of MWh to be produced right when it is needed. Through storing energy for later use, the BESS enables adequacy in the market. The lack of adequacy is especially a problem with systems having a large share of renewable electricity production. The BESS can also have an integrated regulation system which compensates for the variability in the power output from the

solar panels. This type of regulation will make sure the solar panels maximizes the financial return on generated power.

Another type of energy storage is a pumped-hydroelectric technology. This solution makes use of water's potential energy as energy storage. During low demand hours, for example at night, the water will be pumped to a higher elevation. During higher demand hours, as electricity demand rises, the water is released to flow through a turbine that generates electricity. The pump can for example be powered by solar panels during the day [13].

2) *Example:* In the case where the power plant in the city is a solar park, a BESS can be in great use. Most residential consumers' electricity demand peaks in the mornings and in the afternoon, basically before going to-, and after getting home from work. The top hours for solar energy are not during peak demand hours, they are in the middle of the day [14]. The BESS could delay some of that energy being produced during the day to the afternoon when demand is higher. The water pump would have the same application in the example. The BESS would also have a role in making the power output from the solar park steady. A steady power output ensures steady electricity output for all consumers.

3) *Pros:* As described in the introduction, renewable electricity production in the forms of solar panels and wind turbines lack attributes of inertia. This means that there is no ability for the generation spot to reserve energy for later use without the use of separate energy storage. Both batteries and water pumps are managing a time shift in the production of renewable energy, which contributes to them being more reliable. The battery systems are relatively new technology and they are expected to develop further to be more cost effective and with less energy losses [12].

4) *Cons:* Both battery systems and pumped storage plants require high initial costs [15] [16]. This is the main issue with both technologies.

E. Reliability options

1) *Theory:* Reliability options are contracts between the system operator and the producers. The producers sell contracts on specific amounts of capacity to the system operator. If they can't deliver this capacity, they will be penalized. These contracts guarantee the consumers a specific amount of capacity with a price cap on the market price. When producers sell this reliability option, they commit to supplying energy and also to return the extra revenues they obtain when the market electricity price (spot price) is higher than the strike price. For committing to this, the producers get a compensation per produced capacity. The strike price is set in the contract and is set higher than the market price during normal market conditions, which covers the marginal cost for the production. If the producer does not supply, it will either way have to pay the system operator the difference between the spot price and the strike price. That way, the more reliable the producer is, the more it will get paid by the system operator. This counteracts producers to produce less in order to benefit from scarcity pricing that can be very high, meaning that the price per unit produced energy is very high and makes it more profitable to

produce less (and therefore have less expenses from producing for a non renewable power plant) but still make more total profit since the price per MWh gets much more percentage higher than the percentage less production [17] [18] [19].

2) *Example:* The plant in the example has a maximum capacity of 10 MW and wants to sell a reliability option consisting of 9 MW to the system operator. The strike price is agreed to be 800 €/MW. In cases when the spot price is higher than 800 €/MW, for example 900 €/MW, the plant will have to pay the system operator the difference in price for every MW they produce. If the plant produces 9 MW during the period when the spot price is 900 €, the plant will have an income of $9 \text{ MW} * 900 \text{ €/MW} = 8100 \text{ €}$. Since it is in the contract that the power plant has to return the extra revenues, the plant will have an expense to the system operator to the value of $9 \text{ MW} * (900 \text{ €/MW} - 800 \text{ €/MW}) = 900 \text{ €}$. The result for the plant will hence be $8100 \text{ €} - 900 \text{ €} = 7200 \text{ €}$ for the 9 MW it produced, resulting in 800 €/MW. If the power plant instead produces more, for example 10 MW during a time where the spot price is 900 €/MW, the income will be $10 \text{ MW} * 900 \text{ €/MW} = 9000 \text{ €}$ but the expense will still be $9 \text{ MW} * (900 \text{ €/MW} - 800 \text{ €/MW}) = 900 \text{ €}$, since it is 9 MW production that is in the contract. This results in $9000 \text{ €} - 900 \text{ €} = 8100 \text{ €}$ for the 10 MW produced, resulting in 810 €/MW. If the plant does not produce the 9 MW agreed on, but only 8 MW during a period when the spot price is (still) 900 €/MW the income will be $8 \text{ MW} * 900 \text{ €/MW} = 7200 \text{ €}$ and the expense still $9 \text{ MW} * (900 \text{ €/MW} - 800 \text{ €/MW}) = 900 \text{ €}$ resulting in $7200 \text{ €} - 900 \text{ €} = 6300 \text{ €}$ meaning 700 €/MW. The reliability option ensures that the producer is financially rewarded if it delivers a capacity amount exceeding that of the amount agreed upon in the contract. Opposite, the producer is penalized to under-deliver on capacity. This is a good result for the consumers since it makes the electricity market more reliable. The producer in turn benefits from an upfront payment for selling the contract to the system operator, which means it can sell it for as high as it thinks agreeing to this is worth.

[4].

3) *Pros:* Reliability options minimize the risks of producers wanting to withdraw capacity in order to benefit from scarcity pricing. This makes the electricity market more reliable both regarding price and capacity

4) *Cons:* Although reliability options can effectively deliver on ensuring available capacity during periods of scarce electricity supply, these markets are keen to market distortion outcomes. In order for reliability options to work, regulatory authorities are responsible for setting certain parameters required in the implementation. This is a complex task, exposed to increased risk due to market power. This has been noted in the Irish capacity market where authorities failed to do this properly. Consequently, these well-intended, but complex, parameters have led to some market distortions. This in turn can entail regulatory risk [20].

F. Demand elasticity

1) *Theory:* Demand elasticity is, although quite different from other proposals in this study, also a realistic solution

to the adequacy problem. This technology is becoming more relevant as consumer's ability to adapt demand to fluctuations in spot-market prices is seeing great improvement in new products entering the market. Demand elasticity can be improved in many different ways and at great scale, and is therefore relevant for discussion. In a way it is a counter-alternative to implementing a capacity market, since one of the utilities of a capacity market is to mitigate any problems caused by demand-side flaws. Demand elasticity is a solution that recognizes a fundamental flaw of the electricity markets that is highly price-inelastic demand. This fuels high energy prices and contributes to blackouts when electricity supply is scarce. Through increasing demand elasticity enough that the spot price never exceeds the value of electricity to the average consumer, the adequacy problem could be solved. This design feature would theoretically make the electricity system resilient to blackouts. Although this is only true if all demand is truly flexible, in reality, this is unlikely to be achieved. Even though it is expected to see demand elasticity improve during the coming years, a separate capacity market might still be relevant to mitigate other risks with the current electricity market [4].

2) *Example:* Demand elasticity applied to the example would mean that mostly the resident's consumption of electricity is adapted to low demand times. For example, all residents in the village could use washing machines that choose to operate when the demand is at its lowest. This will lead to a slighter less high demand during the top demand times. When using more and more smart technology products, the demand can be shifted more and more.

3) *Pros:* Demand elasticity takes away some of the requirements of available capacity from the producer's side. This will make it easier to balance momentarily produced capacity with momentarily consumed capacity.

4) *Cons:* A lot of responsibility lies on the consumers, that they buy and use these smart technologies. Products with this smart technology are generally new devices that not all customers can afford. There are also limitations on how much electricity consumption can be shifted. Although demand elasticity can somewhat reduce market power, the missing money problem is expected to remain.

G. Strategic reserve

1) *Theory:* Another approach to solving the adequacy problem is through using strategic reserves. In this approach, the system operator purchases or contracts expired or old generators and maintains them for use when electricity generation becomes scarce. The system operator calculates the amount of capacity that needs to be procured. The procurement is paid by the system operator to the power plant upfront. It is important to limit the strategic reserves' interference with the electricity market. If this is not accounted for, the strategic reserve might take revenue from existing plants, since this would lead to lower investments in additional capacity. Therefore, they are only active when all other generators are running. This type of market is easy to implement but comes with the risk of market price manipulation [4].

The strategic reserves plants are chosen by the system operator. Generally plants with high variable costs are chosen, since the strategic reserve is the power plant that will be used the least in the system. With a strategic reserve, the system operator will ensure that the power plant is able to supply the grid with a certain amount of capacity. The power plant is then activated as the electricity price exceeds a set reserve price. The reserve price is decided by the system operator and set so that it exceeds the plant's marginal cost of generation but stays below the VoLL. This is economically possible for the plant since it is the power plant that sells this contract to the system operator [21].

2) *Example:* To implement strategic reserves on the example another power plant will need to be implemented. Let's say the society has a coal power plant that almost perfectly provides to the city. In the summertime though, the citizens use a lot of air conditioning to cool all residential buildings as well as offices etc. On the warmest days of the year, the coal power plant can not provide enough power to meet all demand. The system operator will then write a contract with for example an old oil power plant that can provide with the extra power needed for these irregularly warm days. The system operator will pay the oil plant providing a certain amount of capacity. This contract will ensure that the electricity consumers have enough power at all times.

3) *Pros:* Since the strategic reserve is only activated during the highest demand hours, the use of these expensive plants are minimized. Also, a strategic reserve provides the market directly with the aid of firm capacity. Different from a quantity based approach, which in practice might not be as clear to defining firm capacities. In this case, a strategic reserve option might be advantageous since it is generally thermal plants that offer procured capacity, and for such plants it is elementary to define firm capacity. Another advantage of the strategic reserve, some argue, is that the payment for capacity is only paid to the strategic reserves [4].

4) *Cons:* The strategic reserve generally leaves to the market operator to determine the location of the strategic reserve as well as mix of generation. Due to a more centralized market structure, this can in turn reduce the markets influence to decide generation mix, causing risks in if the types of electricity production are eligible. This does not have to imply that the outcome is bad, since a system operator can do a great job in procuring capacity. But market power certainly increases the risks for under- and over investments in capacity as well as risks suppressing investments in new capacity. Another limit of the strategic reserve is that if the reserve power is offered at a price based on the last commercial spot offer, this removes all scarcity rents potentially precluding investments in new capacity. Additionally, one problem with current strategic reserves is how demand response should be managed. If the strategic reserve isn't allowed to submit bids in real time this will lead to an non-optimal market outcome and dispatch will be inefficient.

5) *Implementation in Sweden:* Generally, Sweden has great electricity production resources. During extra high demand, the demand can often be met with the help of importing capacity on the day ahead market. This means that if the

system operator in Sweden, called "Svenska kraftnät" (SvK) thinks there will be an imbalance in the Swedish market the next day, they can place an order to import capacity from neighboring countries. If the imported capacity is not enough, SvK will activate the strategic reserve [22]. In Sweden there is a law from 2003 that says that the system operator SvK is responsible for that there exists a strategic reserve [23]. Since then, the strategic reserve has had some restrictions regarding how much capacity the strategic reserve can supply. The maximum amount of capacity was 2000 MW in 2003 and in 2017 it was 750 MW [23] [24]. Since the beginning of 2020 an EU reform was implemented that made it harder for SvK to make new contracts with new strategic reserves [25].

The power plants that sign the strategic reserve contract get a fixed cost for agreeing to be a strategic reserve. If the power plant is needed as a reserve, the price for its delivered electricity will be decided on the day ahead market. The price is based on the highest bid for the spot market electricity price [25]. The strategic reserve for the years 2020-2025 is agreed to be the oil power plant Karlshamnverket with an agreement on 562 MW [26]. The owner of Karlshamnverket, Uniper writes on their webpage that the maximum power capacity of Karlshamnverket is 662 MW. Sweden in general, produce a lot of power and exports a lot of power. In the cold wintertime though, it can in some specific hours not be enough. For these situations, when the need for electricity is very high, Karlshamnverket exists. That is why the contract between the oil power plant and SvK is only negotiated during the winter period, from the 16th of November to the 15th of March [27] [22].

6) *Pros:* According to SvK, [22] Sweden has never had to cut off electricity consumption due to lack of capacity for the normal user. In some cases SvK has reduced capacity to certain industries that are contracted as a part of the strategic reserve. This reduction of consumption is the last resort solution if there is lack of capacity. The fact that the normal consumer never has had to cut back electricity usage due to lack of capacity in the grid, means that the strategic reserve works well to ensure capacity to the consumers.

7) *Cons:* The future of strategic reserves in the EU does not look that bright. Since the contract of the strategic reserve expires in the beginning of the year 2025. Like SvK [25] writes on their web page, they are not allowed to contract new strategic reserves after the winter year 2024/2025. The law regulating this is a EU regulation implemented in the year 2019 that says no more new strategic reserves can be contracted. But one can question how economically efficient the strategic reserve solves the adequacy problem. [4] argues that the strategic reserve design in Sweden is an inefficient market solution as the strategic reserve removes all scarcity rents. This means that the signal for electricity shortage is removed which leads to lack of incentives for new investment in production [28].

H. Comparison

The different solutions presented above are summarized in table I. An important note is that the pros and cons for the

TABLE I
COMPARISON OF SOLUTIONS TO THE ADEQUACY PROBLEM

Solution to Adequacy Problem	Theory	Pros	Cons
Regulated price cap	The system operator sets a price cap on electricity.	A regulated price cap can lower the price on electricity and reduce the risks in the market.	A price cap set to low will result in blackouts, a price cap set to high can be politically hard to justify.
Operating reserve	The system operator contracts power with power plants to motivate the plants to meet demand instead of benefiting from scarcity pricing.	Efficient in solving the adequacy problem when the demand is expected to be relatively constant.	Unable to react to shifts in demand as the design is based on predetermined terms.
Energy Storage	Having energy storage as a integrated part of the grid, which introduces inertia to renewables like wind- and solar power.	Energy storage makes unreliable power production reliable. A energy storage system implemented in the grid enables a power production consisting of 100% of wind- and solar power.	High initial cost to implement.
Reliability options	Contract between system operator and producers which motivates the producers to be more reliable and produce more power.	The electricity market becomes more reliable regarding price and supply.	Can be difficult to implement, somewhat dependent on centralized authority and exploited to market power.
Demand elasticity	Electricity consumers use electricity when the demand is at its lowest with the help of technology.	The balance between demand and supply becomes easier.	The responsibility lays on the consumers to buy this kind of technology, which often can result in higher initial costs.
Strategic reserve	A power plant in the system is only used as a reserve and therefore only activated when needed according to the system operator.	Straight forward and relatively quick to implement.	Dependent on centralized authority and exploited to market power.

different solutions will be more or less valuable in different energy systems. For example, a solution with high initial costs can be more or less problematic in different parts of the world with different socioeconomic conditions. With a regulated price cap, the system operator basically chooses between low electricity prices or reliable electricity, whatever they think is most important. Demand elasticity also has a cost perspective which makes it more or less suitable in different socioeconomic conditions. With demand elasticity the cost involves the consumer buying and using a certain type of technology, which can lower their electricity bill. Energy storage and demand elasticity is similar in that way, both solutions involve an investment in a technology that can in the energy storage solution make the supply more reliable, and in the demand elasticity solution make the demand more reactive to price variations. Operating reserve and reliability options both rely on the system operator to motivate them to produce and not withdraw production to benefit from scarcity pricing. An operating reserve is paid upfront to meet certain operating criteria. This is done by contracting with sidelined dispatchable generators obliged to provide generation during certain demand-supply conditions. Reliability options on the other hand are contracted in a way that penalizes generators that do not deliver on procured capacity. In other words, not delivering on the capacity agreement between the system operator and producer results in an economic loss for the producer. With that being said, each solution has its own pros and cons to solve the adequacy problem, different solutions fit better in different electricity systems.

V. SIMULATION

A. Method

1) *Polynomial fitting*: The aim of the simulation was initially to estimate how much Karlshamnsverket, as a strategic reserve, will produce in the year 2024. To do so, data from previous years was needed. The years 2018 through 2021 were chosen. The data set consisted of hourly data extracted from entso-e's webpage, [29]. The data sets only showed data for the days when Karlshamnsverket was producing. Before the data could be used, all days without data points for which

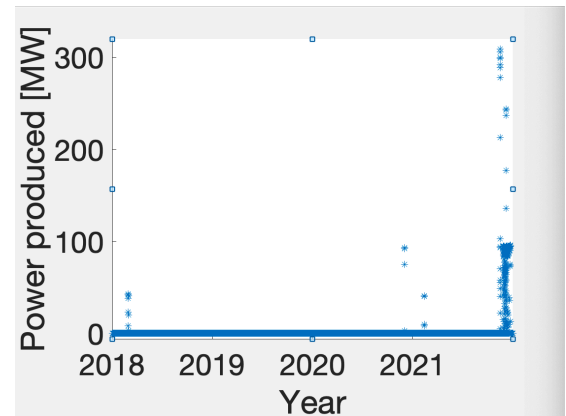


Fig. 1. Capacity produced

Karlshamnsverket was not producing needed to be filled with zeros. Therefore, all hours for the days when SvK did not use

Karlshamnsverket as a strategic reserve, 16th or March until 15th of November were filled with zeros. Doing this,

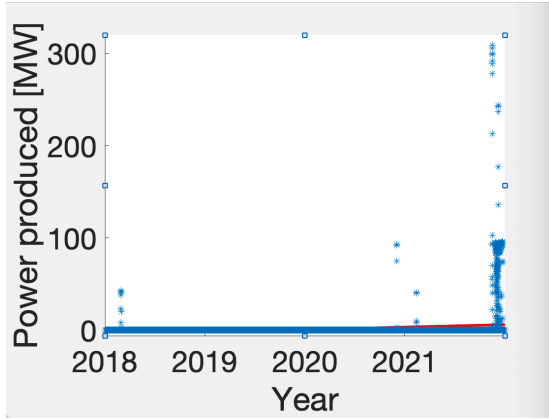


Fig. 2. First degree polynomial with data points

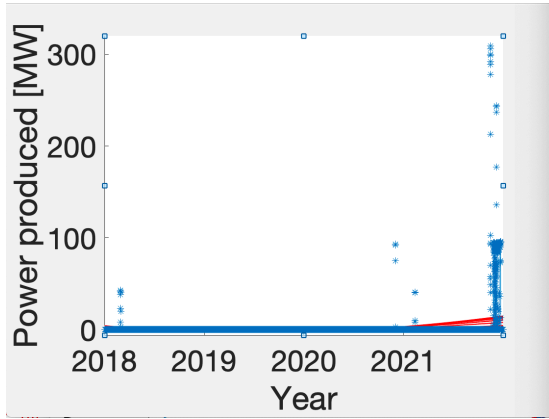


Fig. 3. Second degree polynomial with data points

the data set represents the whole year, but only shows the amount of MW Karlshamnsverket produces because of it being a strategic reserve. The data set was saved as a long vector where every element represented the production in megawatt (MW) for each hour during the years 2018-2021. A plot of this vector can be seen in figure 1. The thick line by the x-axis is all hours where Karlshamnsverket did not produce.

To try and estimate how much power Karlshamnsverket will need to produce as a strategic reserve in the year 2024, different polynomials were fitted to the data vector. No degree of polynomial could fit to the data vector in a satisfying way. To try and make it easier to fit polynomials to the data, another approach to the data vector was used. The data was split up into 24 vectors, each one containing data representing each hour of the day. Since the demand and supply of electricity varies throughout the day, a vector containing the supply from Karlshamnsverket during one hour everyday for four years can be expected to be less fluctuating. Polynomials up to the fourth degree were fitted to each hour's vector data. When using a polynomial of the fourth degree or higher, a warning from MATLAB about the plot could be seen. This way of fitting polynomials didn't either manage to describe the data set in a

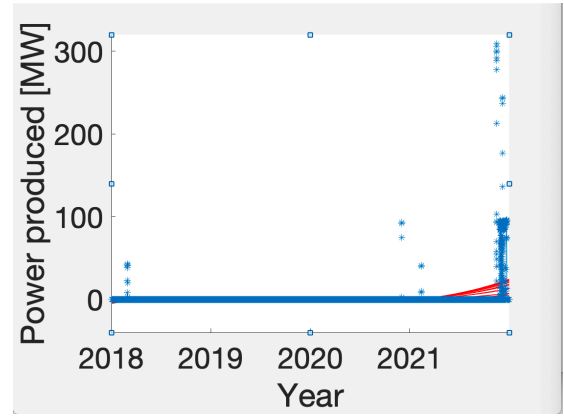


Fig. 4. Third degree polynomial with data points

proper way, like seen in the figures 2, 3 and 4, the 24 datasets where still too varied to be describe with polynomials.

2) *EGARCH equations:* As the polynomial fitting did not work out to estimate the amount of capacity needed from Karlshamnsverket in 2024, EGARCH equations were tried instead. To describe the variation in the data points the app "Econometric modeler" was installed to MATLAB. Within the Econometric modeler there are various kinds of data analyzing models. All different models were applied to the first data set, containing the data for the first hour of every day. The model that could describe the data set the best was the exponential general autoregressive conditional heteroskedastic (EGARCH) model and was therefore the model ud. The measure for the best model fit for the data was the Akaike information criterion (AIC) and Bayes information criterion (BIC) values, which are both criterion used for comparing how well different models fit the data, the lowest AIC- and BIC value is the best model [30]. When using the EGARCH model, one can choose the degree of the GARCH and ARCH to get the best fit. The method to get the best fitted EGARCH equation for each hour was to try different degrees for the GARCH and ARCH, with the constraint that the degree of them should be the same.

The general equation when using the EGARCH model is in the following form:

$$(1 - \lambda_1 L - \lambda_2 L^2 - \dots - \lambda_{nG-1} L^{nG-1} - \lambda_{nG} L^{nG}) \log(\sigma_t^2) = \kappa + (\alpha_1 L + \alpha_2 L^2 + \dots + \alpha_{nA-1} L^{nA-1} + \alpha_{nA} L^{nA}) \left(\left(\frac{|\epsilon_t|}{\sigma_t} \right) - E_h \left(\frac{|\epsilon_t|}{\sigma_t} \right) \right) + (\zeta_1 L + \zeta_2 L^2 + \dots + \zeta_{nA-1} L^{nA-1} + \zeta_{nA} L^{nA}) \left(\frac{\epsilon_t}{\sigma_t} \right) \quad (1)$$

The best fit resulted in AIC and BIC values varying between -1127.1 and -1026.8 for the 11th hour in the day to -210,410 and -210,260 in the 15th hour in the day. In comparison, the AIC value could be as high as 150,000 when using the other models available in the Econometric toolbox, therefore, one can argue that the EGARCH model was the best model for these data sets. The degree of the GARCH is denoted as nG in the equation below, and the ARCH degree is denoted as nA. The degree of the GARCH is denoted as nG in the

equation below, and the ARCH degree is denoted as nA . As a compliment to the EGARCH functions presented in appendix A.

The data points L being used are the amount of capacity produced by Karlshamnsverket in MW. In equation (1), L is the previous data point, L^2 is the data point previous to the previous. The n stands for the number of grades chosen for the specific hour data set. The λ are the GARCH parameters computed by the GARCH model, the α are the ARCH parameters computed by the GARCH model and the ζ are the leverage parameters computed by the GARCH model. The σ are the variances and the ϵ are the errors with the computation. What the EGARCH equation tells us about the different data sets is how the variance, σ is depending on the capacity produced in a certain hour on a certain day. The GARCH model does so by doing estimations based on the previous data points, L and L^2 and so on.

At first, the aim with the EGARCH equations was, as previously stated, to estimate the capacity needed from Karlshamnsverket in the year 2024. This did not succeed, and therefore the EGARCH equation was instead tried to be used to describe the variation of the production from Karlshamnsverket. When looking at the EGARCH equations in the table in appendix A, one can see that the lowest grade GARCH function, which is the fifth grade, corresponds to hour 17, a high demand hour. It can also be seen that the grade of the GARCH model varies more during daytime than during nighttime. Furthermore one can see that the EGARCH model is generally a better fit (the lowest AIC- and BIC values) for the hours during the day when the most electricity is consumed. The possible relation between lower demand hours and the value of the AIC- and BIC values is vague and does not tell much about Karlshamnsverket. Because of this we also did different box-plots with the data using MATLAB. In the box-plot one can see the difference between different hours during the day more clearly.

3) *Box-plots*: Because the polynomials and the EGARCH equations could not help us estimate the capacity needed from Karlshamnsverket in 2024, and the EGARCH equations did not contribute to understanding of the variation in production by Karlshamnsverket, another simulation approach was needed, and therefore, with the help of MATLAB, two different kinds of box-plots was created. The different kinds of box-plots were based on different hours during the day, seen in figure 6, and on different years, seen in figure 5.

B. Results

The results from the different box-plots in figure 5 and 6 show that there is significant deviation from the mean MW value of capacity produced at the strategic reserve. This shows that the need for capacity is rare but significant relative to the median when it is provided. Interpreting the results, this would align well with the requirements on a reliable electricity system, where blackouts are considered a rare event occurring one to a few times a decade. From the box-plots it is seen that the maximum capacity that is sometimes demanded from the reserve plant deviates greatly from the median value. The

box representing the upper- and lower quartile of the data normally seen in a box-plot can not be seen in figure 5 and 6. This is because in an absolute majority of the hours during these years, Karlshamnsverket was not activated. The very most common value being in the data sets was zero. Every hour during which Karlshamnsverket's production was not zero can be seen as a dot in the box-plots.

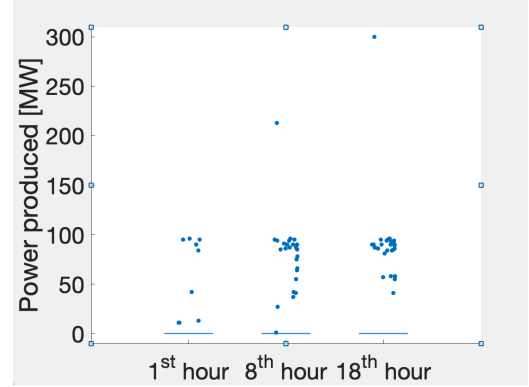


Fig. 5. Box-plots for the 1st-, 8th- and 18th hour

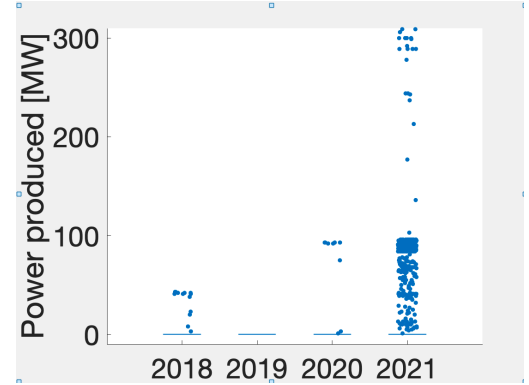


Fig. 6. Box-plots for all hours throughout the years

The box-plots in figure 5 results show that different hours of the day demand more or less sidelined capacity. The least amount of non-zero data points are at the 1st hour, which means that this hour is comparatively the lowest demand hour for capacity out of the three situations. The amount of non-zero data points in the 8th and 18th hours are more, with more or less the same quantity of points. Thus, the demand for capacity is about as frequent at 8 AM as 6 PM. Although the frequency is similar, the spread of data points is greater in the 8th hour. The power demand for the 18th hour weighs more to higher amounts of MW compared to the demand at the 8th hours. This shows that the amount of capacity tends to be greater at the 18th hour compared to the 8th hour. This is what is expected since a strategic reserve gets activated by the system operator when electricity becomes scarce. During typical high demand hours like 8 AM and 6 PM, Karlshamnsverket is needed more often and during a typical low demand hour like

1 AM, Karlshamnsverket is needed less often.

Results in figure 6 show that the need for a capacity reserve is bound to certain years. In 2018 it can be seen that there were some hours with production from Karlshamnsverket. During 2021, a record breaking year, there were substantially more hours during which Karlshamnsverket was producing. In 2021 it can also be seen that the amount of power produced was substantially more in comparison with the three previous years. In general, the utility for Karlshamnsverket seems to have increased over the past few years, with the exception of 2019.

An important disclaimer is that the simulation results regard all provided MW out of the strategic reserve power plant. The production output is not always done with the intent to cover for insufficient generation in the electricity grid. This specific event, which is what the procurement from SvK regards, is much rarer and happens generally once with years in between. Beyond being used as firm capacity, the strategic reserve power plant is used for frequency control. Nevertheless, the results show that the need for reserve power is rare in the Swedish power system, but that when it is needed, the demand can be high and vary greatly from the median output.

VI. DISCUSSION

Different capacity market designs have obviously different pros and cons, as presented in the "Different solutions to the adequacy problem". There does not seem to be a "one that beats all" approach. Instead, different capacity markets incentivize adequate generation in different ways. Some approaches are economically more efficient whilst others might be easier to implement successfully. Another aspect of capacity market design is the generation mix. Depending on the electricity generation resources, a certain capacity market might be more fitting than otherwise. Quite likely, most modern electricity markets would benefit from some type of a capacity market in order to become more reliable and efficient. For the Swedish capacity market the results show that the utility of Karlshamnsverket has increased during the last four years.

The results from the box-plots show volatility in capacity reserve demand. Considering the ongoing shift to renewable electricity production in the whole system, the results might emphasize the need for a different capacity market design in Sweden in the future. It can be argued that Karlshamnsverket at some point won't be able to provide enough installed capacity to assist a wind-turbine heavy electricity grid. Therefore it could be relevant to consider market alternatives that do not leave to the system operator to decide the generation unit. The box-plot results also show that significant demand for reserve capacity is rare. This can substantiate that market conditions need to account for extreme scenarios where electricity demand is high and power generation is not enough to satisfy consumer's needs. Furthermore, since there are goals for Sweden to become fossil free by 2045, an oil powered reserve like Karlshamnsverket might not be relevant. It might be suitable to implement a design that procures firm capacity in another way that a strategic reserve does. In this way, for example using reliability options, energy storage or an operating reserve, adequate capacity in the system can be

reached through market mechanisms and not solely dependent on a decision by the system operator. This way, especially in the case of an operating reserve or energy storage, can secure capacity that is more fitting for a decarbonization narrative if this is an attribute that the market values.

The results shown in the box-plots indicate that the need for a strategic reserve as a hedge against blackouts does not occur every year. Furthermore it shows that the output from Karlshamnsverket varies substantially between different years. The output also varies as the demand for electricity varies throughout the day. These results show the difficulty of projecting future capacity needs from Karlshamnsverket. Since it is essentially only in 2021 the capacity need stood out, this indicates a steep demand curve of capacity, which might not be the case considering capacity reserved peak in demand with years in between. This issue was experienced in the making of this project and as a result found no efficient way to project future capacity needs.

The Swedish strategic reserve market seems to procure enough firm capacity to complement the current electricity system and prevent blackouts from occurring regularly. But one can question how economically efficient the strategic reserve solves the adequacy problem. [4] resonates that the strategic reserve design in Sweden is an inefficient market solution as the strategic reserve alters the electricity prices away from the "efficient spot market pricing". It is also not clear how the strategic reserve will perform as renewable generation becomes a larger share of the system and production becomes more intermittent. Furthermore, SvK's contract with the Swedish strategic reserve plant expires in 2025 and with the changing market conditions it demands for some substitute for procurement of capacity. Additionally, the cost of procured capacity in Sweden indicates that the cost has seen a steady increase during the last years, and indicates that the cost will become even more expensive in the future. It might be relevant to look for capacity market alternatives to ensure that adequate generation investments are being made in the future power system.

VII. CONCLUSION

Capacity markets incentivize investments in capacity, different capacity market designs do this in different ways. Investments in Capacity means a more reliable market. The capacity market design used in Sweden, strategic reserve is able to produce enough capacity to complement the current electricity system and prevent blackouts. To conclude, considering the changing nature of electricity markets, it is important to continue ensuring that investments in adequate capacity are being made.

APPENDIX A

TABLE WITH EGARCH EQUATIONS

ACKNOWLEDGMENT

The authors would like to emphasize great gratitude to our supervisor Mohammad Reza Hesamzadeh (Division of Electric Power and Energy Systems, KTH). His expertise, knowledge and frequent input has been of great use and help during the making of this study.

REFERENCES

- [1] A. K. Erenoğlu, O. Erdinç, and A. Taşcıkaraoğlu, *Pathways to a Smarter Power System*. London, England: Academic Press, 2019, pp. 1–27.
- [2] J. Vuckovic, “Nikola tesla: the man time forgot,” *IEEE Potentials*, vol. 9, no. 3, pp. 53–54, 1990.
- [3] M. Hesamzadeh and D. Biggar, *The Economics of Electricity Markets*, ser. IEEE Press. Noida, India: Wiley, 2014, pp. 130–137.
- [4] P. Cramton, A. Ockenfels, and S. Stoft, “Capacity market fundamentals,” *Economics of Energy Environmental Policy*, vol. 2, no. 2, pp. 27–46, 2013.
- [5] International Energy Agency, “Global energy review 2021,” *International Energy Agency*, pp. 6–20, 2021.
- [6] M. Amelin and L. Söder, “Taking credit,” *IEEE Power and Energy Magazine*, vol. 8, no. 5, pp. 47–52, 2010.
- [7] L. Söder, “Analysis of pricing and volumes in selective capacity markets,” *IEEE Transactions on Power Systems*, vol. 25, no. 3, pp. 1415–1422, 2010.
- [8] P. Simshauser, “Vertical integration, credit ratings and retail price settings in energy-only markets: Navigating the resource adequacy problem,” *Energy Policy*, vol. 38, no. 11, pp. 7427–7441, 2010.
- [9] P. Holmberg and T. P. Tangerås, “Strategic reserves versus market-wide capacity mechanisms,” *IFN Working Paper*, no. 1387, pp. 2–19, 2021.
- [10] J. Wang, X. Wang, and Y. Wu, “Operating reserve model in the power market,” *IEEE Transactions on Power Systems*, vol. 20, no. 1, pp. 223–229, 2005.
- [11] U. Helman, B. F. Hobbs, and R. P. O’Neill, *Competitive Electricity Markets*, ser. Elsevier Global Energy Policy and Economics Series. Oxford, England: Elsevier, 2008, pp. 179–191.
- [12] J. Figgenger, P. Stenzel, K.-P. Kairies, J. Linßen, D. Haberschusz, O. Wessels, G. Angenendt, M. Robinius, D. Stolten, and D. U. Sauer, “The development of stationary battery storage systems in germany a market review,” *Journal of Energy Storage*, vol. 29, p. 101153, 2020.
- [13] C. A. Hill, M. C. Such, D. Chen, J. Gonzalez, and W. M. Grady, “Battery energy storage for enabling integration of distributed solar power generation,” *IEEE Transactions on Smart Grid*, vol. 3, no. 2, pp. 850–857, 2012.
- [14] Svenska Kraftnät. (2022, Mar.) Elstatik. [Online]. Available: <https://www.svk.se/om-kraftsystemet/kraftsystemdata/elstatistik/>
- [15] M. T. Lawder, B. Suthar, P. W. C. Northrop, S. De, C. M. Hoff, O. Leitemann, M. L. Crow, S. Santhanagopalan, and V. R. Subramanian, “Battery energy storage system (bess) and battery management system (bms) for grid-scale applications,” *Proceedings of the IEEE*, vol. 102, no. 6, pp. 1014–1030, 2014.
- [16] F. A. Canales, A. Beluco, and C. A. B. Mendes, “A comparative study of a wind hydro hybrid system with water storage capacity: Conventional reservoir or pumped storage plant?” *Journal of Energy Storage*, vol. 4, pp. 96–105, 2015.
- [17] C. Batlle, C. Vázquez, M. Rivier, and I. J. Pérez-Arriaga, “Enhancing power supply adequacy in spain: Migrating from capacity payments to reliability options,” *Energy Policy*, vol. 35, no. 9, pp. 4545–4554, 2007.
- [18] M. Bidwell, “Reliability options: A market-oriented approach to long-term adequacy,” *The Electricity Journal*, vol. 18, no. 5, pp. 11–25, 2005.
- [19] L. Andreis, M. Flora, F. Fontini, and T. Vargiolu, “Pricing reliability options under different electricity price regimes,” *Energy Economics*, vol. 87, p. 104705, 2020.
- [20] P. C. Bhagwat and L. Meeus, “Reliability options: Can they deliver on their promises?” *The Electricity Journal*, vol. 32, no. 10, p. 106667, 2019.
- [21] P. C. Bhagwat, J. C. Richstein, E. J. Chappin, K. K. Iychettira, and L. J. De Vries, “Cross-border effects of capacity mechanisms in inter-connected power systems,” *Utilities Policy*, vol. 46, pp. 33–47, 2017.
- [22] Svenska Kraftnät, “Kraftbalansen på den svenska elmarknaden, rapport 2020,” *En rapport till Infrastrukturdepartementet. Ärendenummer: 2020*, vol. 334, pp. 19–22, 2020.
- [23] Riksdagen. (2022, Mar.) Lag (2003:436) om effektreserv. [Online]. Available: https://www.riksdagen.se/sv/dokument-lagar/dokument/svensk-forfattningssamling/lag-2003436-om-effektreserv_sfs-2003-436
- [24] ——. (2022, Mar.) Förordning (2016:423) om effektreserv. [Online]. Available: https://www.riksdagen.se/sv/dokument-lagar/dokument/svensk-forfattningssamling/forordning-2016423-om-effektreserv_sfs-2016-423
- [25] Svenska Kraftnät. (2022, Mar.) Effektreserv. [Online]. Available: <https://www.svk.se/aktorsportalen/systemdrift-elmarknad/information-om-stodtjanster/effektreserv/>
- [26] ——. (2022, Mar.) Effektreserven för 2020-2025. [Online]. Available: <https://www.svk.se/aktorsportalen/systemdrift-elmarknad/information-om-stodtjanster/effektreserv/effektreserven-for-2020-2025/>
- [27] Uniper. (2022, Mar.) Effektreserven är livlinan i det svenska elsystemet. [Online]. Available: <https://www.uniper.energy/sverige/reservkraft/karlshamnsverket>
- [28] U. Helman, B. F. Hobbs, and R. P. O’Neill, *Competitive Electricity Markets*, ser. Elsevier Global Energy Policy and Economics Series. Oxford, England: Elsevier, 2008, p. 333.
- [29] ENTSO-E. (2022, Apr.) Actual generation per generation unit. [Online]. Available: <https://transparency.entsoe.eu/generation/r2/actualGenerationPerGenerationUnit/show>
- [30] D. Lord, X. Qin, and S. R. Geedipally, *Highway Safety Analytics and Modeling*. Chennai, India: Elsevier, 2021, pp. 17–57.

Modeling of hydropower in Spine - Optimizing Electricity Production With a Piece-Wise Linear Dependency

Siri Löfgren and Iris Seppälä

Abstract—Hydropower plays an important role in the Swedish power system and is a valuable renewable energy source with great ability for regulation. It is, therefore, crucial to plan and optimize hydropower in a way that is effective. In this project, the Skellefte River is modeled with the software Spine. The focus is on applying a piece-wise linear function to describe the electricity production, instead of a simpler linear one, and optimizing the profit. The results of the optimization indicate that the piece-wise linear function gives accurate values on the electricity production. This work has also further contributed to the development of Spine.

Sammanfattning—Vattenkraft spelar en viktig roll i det svenska elsystemet och är en värdefull förnybar energikälla med stor regleringsförmåga. Det är därför avgörande att planera och optimera vattenkraft på ett effektivt sätt. I detta projekt modelleras Skellefteälven med programvaran Spine. Fokus ligger på att tillämpa en styckvis linjär funktion för att beskriva elproduktionen istället för att använda en enklare linjär funktion. Modellen optimeras efter pris. Resultaten av optimeringen indikerar att den styckvis linjära funktionen ger korrekta värden på elproduktionen. Detta arbete har också bidragit till den fortsatta utvecklingen av Spine.

Index Terms—hydropower, Spine, optimization, modeling, piece-wise linear function

Supervisors: Mikael Amelin

TRITA number: TRITA-EECS-EX-2022:147

Symbol	Unit	Description
\hat{Q}	HE	Maximum discharge
\bar{Q}	HE	75 percent of maximum discharge
\hat{H}	MW	Installed effect
\bar{H}	MW	Electricity production which corresponds to 75 percent of maximum discharge
\hat{M}	HE	Maximum reservoirs content
M_{start}	HE	Reservoirs content in the beginning
M_{end}	HE	Reservoirs content in the end
V	HE	Local inflow
R_s	min	Delay time for spillage
R_q	min	Delay time for discharge
μ_1	MWh/HE	Marginal production equivalent in segment 1
μ_2	MWh/HE	Marginal production equivalent in segment 2

I. INTRODUCTION

Many of today's local and global environmental problems are connected to society's energy sector and above all to the usage of fossil fuels. In 2016, as much as 73.2 percent of all global greenhouse gas emissions came from the energy sector [1]. According to IPCC the energy demand is expected to increase in many sectors around the world and

if conventional energy sources keep being used, the negative effects on the environment continue to worsen [2]. It is necessary to transition to renewable energy sources to obtain an ecologically sustainable society, which is predominantly dependent on improving technology to increase the energy efficiency and other alternative solutions [3]. For instance, the ongoing electrification of the steel industry by utilizing hydrogen, and the electrification of the transport sector both need clean renewable energy to be considered sustainable and to not contribute further to climate change [4], [5].

Today, more than ever, the importance of energy independence has also become clear. With the ongoing crisis in Europe, there is a strong will to reduce the dependency on Russian gas. This means that alternative renewable and local energy sources must be implemented in the energy system and used in an effective way. [6]

The problem with having a lot of renewable energy in a power system is the unbalances it brings to the system. The changes in production and consumption lead to changed grid frequency, and if not balanced out directly, it leads to blackouts. The wind- and solar power generation, as well as the demand, are not constant. They vary depending on how much sunlight and wind are available and therefore can only be regulated downwards. Hydropower plants, on the contrary, have good regulating ability. This is because they can be planned so that the available water flow can be utilized so that the generation of electricity answers to the demand rather than how much water flows in the river continuously [7]. This makes hydropower a reliable, affordable and sustainable energy source. However, the capacity of the reservoirs is not unlimited and therefore it of interest to find out how well hydropower can contribute to the balancing of the power system with a lot of renewable energy in it.

The goal of this project is to build an existing model of the Skellefte River with the newly developed software Spine and develop it further [8]. Before this project, other similar projects have been done with the software Spine [9]. What differs this project from them is that the aim is to use a piece-wise linear function to describe the electricity production rather than a linear one, in order to get more accurate result.

II. BACKGROUND

A. History of Swedish hydropower and power system

Globally, the energy bound in water has been used for thousands of years. The earliest arrangements were water wheels

and they can be dated back to 4000BC. However, the modern day history of hydropower is often considered to have begun in mid 18th century [10]. At the beginning of the 20th century, the interest in hydro power grew in Sweden. This period is also called "The era of regional hydropower" and it took place from 1905 to 1935. At this point, many power companies were established and hydropower plants were built to provide the industries and power stations with electricity [11]. It was also at this time alternating current began to replace direct current which made it possible to transfer electric energy over long distances [7]. Around 1960, substantial parts of Sweden's hydropower resources were expanded, this is also when cheap accessible electricity began to be considered a social right in Sweden [11].

Today almost 45 percent of the electricity production in Sweden is generated by hydropower [12]. It is not only in Sweden but globally that hydropower is a major renewable energy source. According to [10] it is estimated that there are at least 11 000 hydropower stations and 27 000 generating units in the world. In 2019 the total installed capacity in the world was 1.308 GW which makes hydropower the worlds largest source of renewable energy [2]. Therefore it is of high relevance that this energy source is used in a smart efficient way.

B. Hydropower and key concepts

The basic principle of hydropower generation is based on converting the potential energy stored in water, to mechanical energy, by using turbines. The turbine drives a generator which converts the mechanical energy to electric energy. The water level differences and flows are necessary for building of hydropower plant, hence building a dam is often needed. The water is led through a gate, a hatch, and a tunnel to the turbine. This is what gives hydropower its regulating ability, the hatches can be opened and closed based on the electricity demand. In other words, the quality that makes hydropower a good regulating source is that it allows us to store water to be used later. It is extremely hard to store electricity but the energy accumulated in water can be stored in reservoirs. [13]

1) *Reservoirs*: In hydropower plants, the pool where the water mass is stored are called reservoirs. They enable a controlled discharge by allowing us to store water to be saved later [14].

2) *Relative efficiency*: The relative efficiency of the hydropower plants is dependent on the discharge through the turbines, as well as the height difference between the water levels of up- and downstream. It tells how much energy is produced for each cubic meter of water that passes through the turbines compared to the maximal amount of energy that can be produced [14].

3) *Spillage*: Spillage is the water mass that does not go through the turbines but is led by it to the next power plant. The spilled water does not contribute to the electricity production. Water can be spilled via the natural riverbed or through special hatches in the power plant. This is sometimes needed when a power plant has a full reservoir. [14].

4) *Discharge*: Discharge is the water mass that is led through the turbines and is therefore contributing to electricity production [15].

5) *Local inflow*: The local inflow consists mainly of water that comes from rainfall and melted snow. Different factors affect the amount of local inflow from an area. For example the size of the drainage area, vegetation and lakes in the surroundings [16].

6) *Marginal production equivalent*: The marginal production equivalent represents the increase in electricity production when there is a small increase in discharge [14].

7) *Water delay time*: The time it takes for the water to flow from the upstream station to the downstream station. [17]

C. Planning and optimizing hydropower

To reach the full potential of hydropower and to get maximum profit, careful planning is needed. The hydropower plants are planned so that the spillage is as small as possible. It is also important to optimize the power plants so that they operate with high relative efficiency, which depends on the amount of discharge. Another factor, that must be taken into account, is electricity prices. When the price is high there should be more water discharged and while the prices are low, it is worth more to store the water and not discharge it [18]. It is useful to optimize the operation by implementing a piece-wise linear dependency with two segments on the production as a function of discharge, this allows us to add an operating point where the generator produces electricity with the highest relative efficiency. This operating point is used when the prices are not extreme. In this model, it is assumed that the highest relative efficiency is at 75 percent of the maximal discharge and that the marginal production equivalent for the second segment is 5 percent lower than for the first. This is a standard assumption which is used for example in [14] and [18]. Generally, the relative efficiency of the power plant decreases with an increasing amount of discharge [14].

D. Skellefte River

The river, which the case study is preformed on, is the Swedish Skellefte River located in Västerbotten. The river's first hydropower plant was built in 1906 in Finnfors [19] and today there are 17 hydropower plants along the river [20]. With its total installed capacity on about 1000 MW, the river makes up for 6 percent of all hydropower generation in Sweden [21]. The 410 km long river has suitable surroundings with a extensive drainage area containing a large number of lakes. These lakes function as reservoirs, which give good opportunities for regulation [21]. In this report 15 of the 17 hydropower plants have been included, leaving out the smallest two power plants which do not contribute much to the production. The data from Skellefte River that has been used for the simulation can be found in Table I and Table II.

TABLE I
SKELLEFTE RIVER DATA

Hydropower plant	$\hat{H}(MW)$	$\hat{Q}(HE)$	$R_{Q(min)}$	$R_{S(min)}$
Sädva	31	70	2880	2880
Rebnis	64	80	2880	2880
Bergnäs	8	160	60	60
Slagnäs	7	160	240	240
Bastusel	100	170	60	60
Grytfors	31	165	15	15
Gallejaur	214	310	30	150
Vargfors	131	320	180	180
Rengård	36	220	180	180
Båtfors	42	280	180	180
Finnfors	54	300	180	180
Granfors	40	240	180	180
Krångfors	62	240	180	180
Selsfors	61	300	180	180
Kvistforsen	130	300	0	0

TABLE II
SKELLEFTE RIVER DATA

Hydropower plant	$\hat{M}(HE)$	$M_{start(HE)}$	$M_{end(HE)}$	$V(HE)$
Sädva	168000	99057,77	93831,11	5,43
Rebnis	205560	70243,51	59524,12	3,68
Bergnäs	216120	1117,2	891,1	22,29
Slagnäs	768	384	537,6	0
Bastusel	8208	5581,44	5417,28	0,258
Grytfors	1248	1060,80	1110,72	3,78
Gallejaur	3600	1224	2808	15,356
Vargfors	4008	3386,76	3847,68	3,558
Rengård	1400	1022	770	10,37
Båtfors	1330	1117,20	891,1	2
Finnfors	300	234	234	0
Granfors	280	232,40	212,8	0
Krångfors	330	201,30	207,9	0
Selsfors	500	40	200	0
Kvistforsen	1120	769,07	560	1,327

III. HYDROPOWER MODELING IN SPINE

A. Spinetoolbox and SpineOpt

Spine is an EU project that started in 2017, with aim to develop an open source software that could be used to plan future European energy grids. Spine Toolbox is a Python package which allows us to define and manage data, as well as provides a framework for modeling complex systems. It allows us to make a visual representation of the workflow. SpineOpt is an energy system modeling package that makes sector-specific modeling with adjustable temporal and stochastic structures possible. [22]

B. Model of the Skellefte River

This model maximizes the total profit of sold electricity over one week. The SpineOpt tool optimizes with respect to start and end levels in reservoirs, limitations of water level

in reservoirs, as well as electricity production. The model takes into account several parameters such as; electricity price, maximal discharge and reservoirs levels, the local inflow of water, and the time delay of water flow. Every power plant in the Skellefte River is constructed with the same principle. As an example, we begin by describing the schematic of the model, which can be seen in Fig. 2, for one of the power plants, namely the Rebnis power plant.

1) *Objects and relationships*: When modeling in Spine the two fundamental concepts are object and relationships. Different objects are being related to one another by different types of relationships. The types of object being used in Spine is further explained below and can be seen in Fig. 1. To construct the model of Rebnis power plant in Spine the objects displayed in table III are being used.



Fig. 1. Symbols in Spine [15]

Units: In this model units represents the power plants and the electricity load.

Nodes: Each power plant has two water nodes an upper node at the entrance of the power plant and a lower node at the exit of the power plant. These represent the water level in the reservoirs. All units are also connected to a common electricity node.

Connections: Represent the water flow between two power plants. There are two connections for each power plant one for the spilled water and one for discharged water.

Commodity: Different types of energy are represented with this object. In this model, there are two commodities: water and electricity.

TABLE III
OBJECT LIST

Unit	Node	Connection	Commodity
Rebnis	Rebnis upper	Spill	Water
Electricity load	Rebnis lower	Discharge	Electricity
	Bergnäs upper		
	Electricity		

The objects are related to one another by relationships. In Spine, there are different types of relationships. The ones being used in the Skellefte River model can be seen in table IV.

TABLE IV
RELATIONSHIP LIST

Relationship class	Description
unit_from_node	A link from a node to a unit
unit_to_node	A link from a unit to a node
unit_node_node	A link from a node to a unit and to another node
connection_to_node	A link from a connection to a node
connection_from_node	A link from a node to a connection
connection_node_node	A link from a node to a connection and to another node
node_commodity	Defines a commodity for a node, a link from a node to a commodity

2) *Parameters*: Parameters specify the behavior or property of the objects and relationships. All parameters used in this schematic model is displayed in Table V.

TABLE V
PARAMETER LIST

Object/ Relationship	Parameters	Description
Upper node	demand	Local inflow of water to node
Upper node	fix_node_state	Start and end water level in node
Upper node	has_state	Tells that node is a reservoir
Upper node	state_coeff	Reservoir efficiency
Upper node	node_state_cap	Maximum water level in node
Electricity node Electricity load	vom_cost	Hourly price of electricity
unit_from_node	unit_capacity	Capacity of hydropower plant both in terms of maximal discharge and installed capacity
unit_node_node	fix_ratio_out_in_unit_flow	Conversion rate of water
connection_from_node	fix_connection_flow	Spill and discharge hours before simulation
connection_node_node	fix_ratio_out_in_connection	Ratio of water flow
connection_node_node	connection_flow_delay	Water delay time
unit_node_mode	unit_incremental_heat_rate	Efficiency of power plant
unit_to_node	operating_point	Breaks down component into segments

3) *Specifying the parameters of the objects*: There are one or more parameters for each object that specify the behaviour of the object. In the Skellefte River model only the upper nodes have specified parameters. `fix_node_state` represents the water levels in the beginning and at the end. The parameter `has_state` tells us if the node is a reservoir. `state_coeff` is the reservoir efficiency, represented by 1 if there are no losses in the reservoir. The maximum level in the reservoir is represented with the parameter `node_state_cap`.

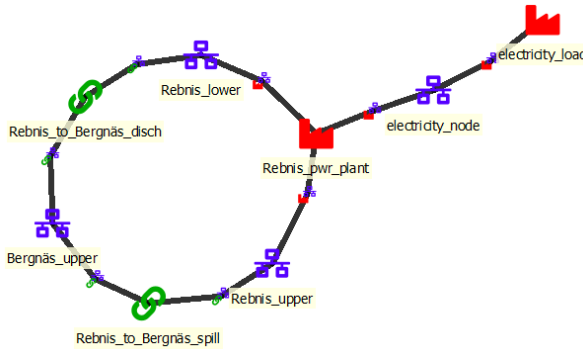


Fig. 2. Schematic of the model

4) *Specifying the parameters of the relationships*: To describe the construction and use of parameters in the relationships we begin at the electricity load that can be found at the far right in Fig. 2 and continue systematically along the figure.

Electricity load to electricity node

The electricity load and node are linked via `unit_from_node` relationship. In this relationship, we have a time series parameter, called `vom_cost`, which tells us hourly electricity prices. The prices are set to positive values.

Rebnis power plant to water nodes and electricity node

Water flows from Rebnis upper node either, through the power plant where it generates electricity to then end up in lower node of the power plant, or it spills past the power plant. The electricity produced goes from the power plant to the electricity node. The following relationships and parameters are used to simulate this. The relationship `unit_from_node` from the Rebnis upper node to the Rebnis power plant is specified with the parameter `unit_capacity`, which describes the maximum discharge in the power plant. The parameter `fix_ratio_out_in_unit_flow` is set on the relationship `unit_node_node` between Rebnis upper and lower node with the value 1.0, since it represents the conversion from water to water between the upper and lower nodes, and no water vanishes.

The `unit_node_node` relationship between Rebnis upper and electricity nodes represents the amount of electricity produced for each unit of water. It is here we want to have the piece-wise linear dependency. For this we use the relationship parameter `unit_incremental_heat_rate` and set the value to be of type Array with two segments. The values we want to put in are the inverses of the marginal production equivalents. We take the inverse due to Spine having specified the values with the unit resource/MWh and our values are defined as MWh/resource. The `unit_incremental_heat_rate` requires operating points to function. We define this parameter, `operating_point`, in the `unit_to_node` relationship between power plant and electricity node. The `operating_points` must be defined as an array type with the corresponding dimensions to the `unit_incremental_heat_rate`. The input values in the array are \dot{H} displayed as a percentage and 1. In addition to this the power plants maximal capacity \dot{H} must be defined somewhere. This is appropriate on the relationship between power plant and electricity node. This is done by using the parameter `unit_capacity`.

Rebnis to Bergnäs spillage and discharge connections

Spill is the water mass the Rebnis power plant cannot utilise to generate electricity. The spilled water goes from the Rebnis upper node directly to the upper node of the next power plant, Bergnäs. Therefore we need a separate connection for the spill to demonstrate that there is a water flow that does not pass through the power plant but goes straight from Rebnis upper node to the next upper node.

As opposed to spillage, the discharge goes through the power plants turbines and generate electricity. After the power plant, the discharge water flows through the lower node of the Rebnis power plant and then, after some time, reaches Bergnäs upper node. The discharge connection demonstrates this separate water flow. Both the spill and discharge connections have the same kind of relationships with the same

parameters. The parameter `fix_connection_flow` is defined on the relationship `connection_from_node`, this is a time series that tells us the average spillage and discharge in the first hours before the simulation. For the relationship `connection_node_node`, that links Rebnis lower node to Bergnäs upper node via the connection, there are two specified parameters. One describing the ratio of the incoming and outgoing water, which is set to one as there is no loss in the connection. By also setting the parameter `connection_flow_delay` on the relationship this tells how long it takes for the water to flow between the nodes. The `connection_flow_delay` is also set between the Rebnis upper and Bergnäs upper nodes to describe the water delay time for the spillage.

Commodity and nodes

Every node in this model is connected to the relationship `node_commodity` to show that the nodes are in balance, the upper and lower nodes of the power plant with water and the electricity node with electricity.

C. Calculations

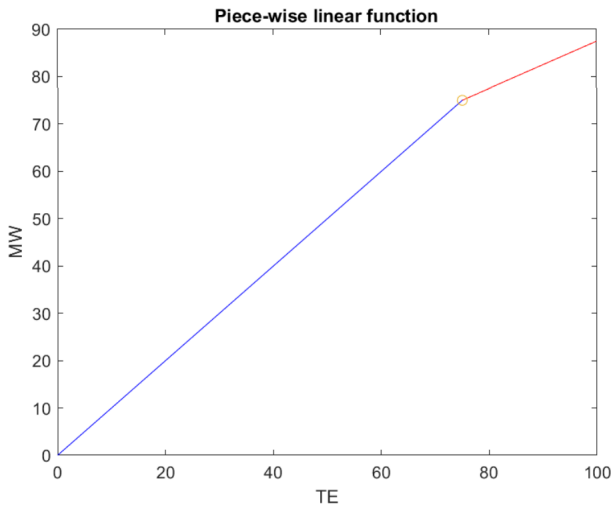


Fig. 3. An illustrating example of a piece-wise linear function with two segments, operating point marked with a circle

To be able to model hydropower, the dependency between the head, the discharge and the electricity production needs to be approximated as some sort of linear function. To increase the precision of the model we approximate the dependency between the electricity production and discharge as a piece-wise linear function with two segments, instead of as a linear function as in earlier works [15], [9] (see Fig. 3). We define the breakpoint to be by the value of discharge where the local efficiency of the power plant is maximal. This leads to that these values of the breakpoints occur more often in the solution to the optimization problem. To accomplish this we need to approximate the marginal production equivalent for each segment for each power plant. For the piece-wise linear function it is assumed that the local maximum for the efficiency is at 75 percent of the maximal discharge. It is also

assumed that the marginal production equivalent for second segment is 5 percent lower than for the first segment. With these assumptions we can construct an equation system of equations 1-3 and solve it for μ_1 and μ_2 . [14]

$$\dot{Q} = 0,75\hat{Q} \quad (1)$$

$$\hat{H} = \mu_1\dot{Q} + \mu_2(\hat{Q} - \dot{Q}) \quad (2)$$

$$\mu_2 = 0,95\mu_1 \quad (3)$$

To be able to define the operating point in Spine, we need to calculate the value of the production that corresponds to the value of discharge (75 percent of the maximum). This can be done by solving the equation 4 for \hat{H} .

$$\hat{H} = \mu_1\dot{Q} \quad (4)$$

TABLE VI
RESULTS OF CALCULATIONS

hydropower plant	$\hat{H}(TE)$	μ_1 (MWh/TE)	μ_2 (MWh/TE)
Sädva	23,524	0,448	0,426
Rebnis	48,607	0,810	0,769
Bergnäs	6,0729	0,051	0,04
Slagnäs	5,3160	0,044	0,0421
Bastusel	75,94	0,595	0,566
Grytfors	23,555	0,190	0,181
Gallejaur	162,59	0,699	0,664
Vargfors	99,502	0,415	0,394
Rengård	27,34	0,1657	0,157
Båtfors	31,900	0,152	0,144
Finnfors	41,013	0,182	0,173
Granfors	30,385	0,169	0,160
Krångfors	47,0884	0,263	0,249
Selsfors	46,2377	0,206	0,196
Kvistforsen	98,771	0,439	0,417

IV. CASE STUDY

To be able to optimize the 15 power plants in Skellefte River, they had to be modeled. This was done with the help of an existing model in Spine [8]. The model optimizes the power plants with respect to electricity prices. After the model was built, a piece-wise dependency on the electricity production was implemented to increase the optimization accuracy. The problem was then solved with the help of SpineOpt.

The data used in this project, specifically installed capacity, maximal discharge, reservoir levels, local inflow and time delay, are real data for the Skellefte River and was provided by our supervisor. The electricity prices used in this project are real prices from an actual week and have been used in previous articles such as [23].

After the first simulations, the results showed that there were a lot of spillage for some hours in some of the power plants. This was due to the there being a fixed value for the end reservoir level, which forced the program to spill water to be

able to satisfy the required reservoir levels. To eliminate this the values of the end reservoir levels were manually iterated to such values that the spillage was reduced to zero on all power plants. The spillage had to be minimized in order to make the model more applicable to reality.

A. Results

After SpineOpt had solved the optimization problem for the model, we plotted the results of the optimization with help of Spine. The red graphs, Fig.4 and Fig. 6, show the electricity production every hour of the week on two of the power plants, namely Kvistförsen and Granfors, when there is unreasonable amount of spillage on the power plants. The spillage was eliminated by manually iterating the fixed end content of the reservoirs to higher values.

The Fig.5 and Fig.7 show the production of electricity every hour of the week on the same two power plants, but with spillage eliminated. It is easy to verify the results being accurate by looking at the dominating values on each figure and compare it to the values of the operating points, the values that should be most common, should be the \hat{H} values, the installed capacity \hat{H} , as well as 0 when the electricity price are not extreme, high, respective low.

In Fig.5 and Fig.7 we can see the electricity production as a function of time. On the y-axis we have the production in MW and on the x-axis the time as dates. If we now would take the power plant Kvistförsen as an example, it shows that the dominating values are 98,73 MW, 130 MW and 0 MW. The \hat{H} at Kvistförsen is 130 MW, the \hat{H} is 98.77 MW, which are the values we want.

One can clearly see by comparing the red and blue graphs on the Fig.4-7, that eliminating the spillage has an impact on the electricity production. For example, when eliminating the spillage, the Granfors power plant is not mainly operating on \hat{H} , but also on the \hat{H} . In other words there are more variation in the electricity production, even though the prices are the same in both cases. The results from the rest of the power plants are presented in the appendix.

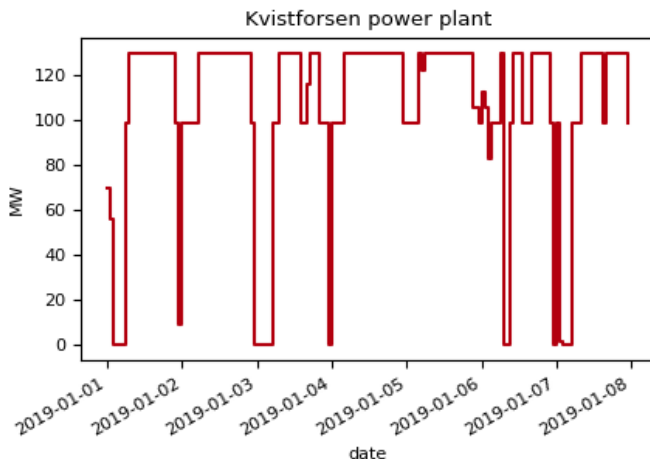


Fig. 4. Electricity production on Kvistförsen power plant with unreasonable spillage

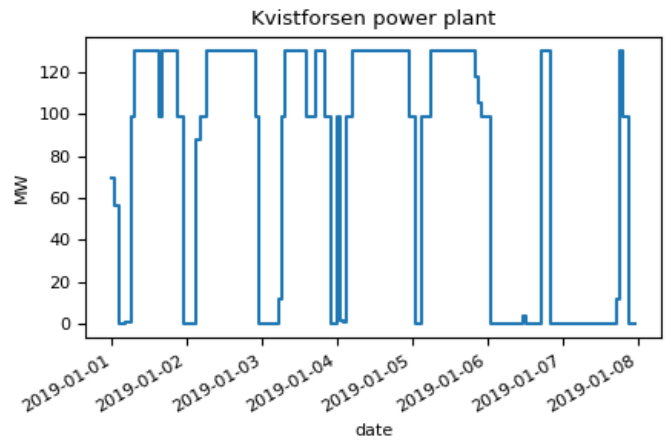


Fig. 5. Electricity production on Kvistförsen power plant with zero spillage

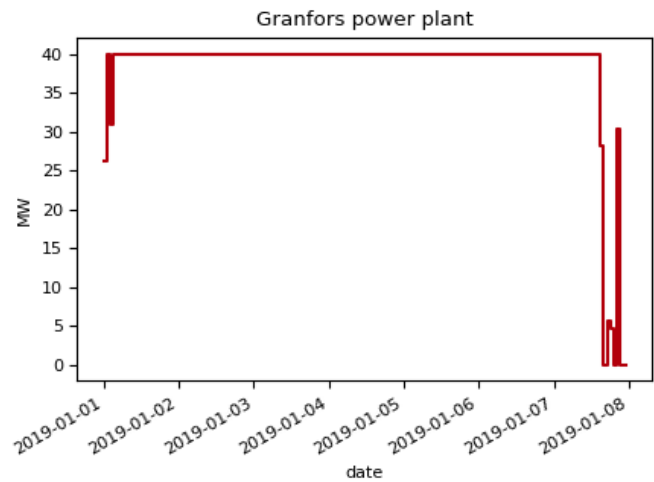


Fig. 6. Electricity production on the Granfors power plant with unreasonable spillage

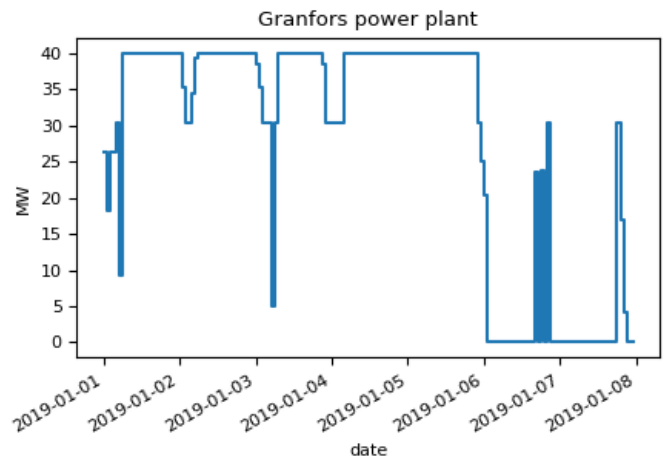


Fig. 7. Electricity production on the Granfors power plant with zero spillage

B. Discussion

The need for energy increases while the world population increases, as well as new technology is being developed and industries being electrified. This means that it is extremely

important to be able to ensure that everyone has access to clean energy, by making sure that the electricity production is sustainable and that electricity prices stay affordable. As mentioned previously, hydropower has a great regulating capacity in comparison to other renewable energy sources. Other energy sources with good regulating abilities are in general fossil based. In order to reduce negative impact on environment and climate from energy production they must be phased out from the energy system. It is therefore of interest to work out just how much regulating capacity there is in hydropower, in other words how much can we rely on its regulation when fossil fuels are not an option. Tools for planning and optimizing energy systems easily, efficiently, and accurately are therefore essential.

Using a piece-wise linear function for the electricity production in Spine gave us results that mostly correspond to the values that we were expecting to get. However, there are some anomalies from the expected values which can be due to several different reasons. The model has a fixed value for the reservoir levels for the last hour. This forces the hydropower plants to spill water, that potentially did not have to be spilled, in order to satisfy the required end level in the reservoirs. Spillage is not desired in general, since it is energy gone wasted. To get even more optimal results, that will say a more energy efficient system, the spillage has to be minimized on each power plant in the model. In other words, the initial model does not recognize the value of saved water. This resulted in there being unreasonable spillage at some hours. Large amount of spillage plays part on the appearance of the graphs as can be seen in Fig.4 and Fig.6.

By minimizing spillage for the power plants in the model, a value for saved water is indirectly set. As can be seen in Fig.4 and Fig.6, the power plants operate more frequently on the installed capacity than in Fig.5 and Fig.7. When setting a higher value for the end reservoir levels, the model is forced to save water to reach this goal. Since the model optimizes with respect to price, the initial model operates on the installed capacity to get as much profit as possible due to high prices. The updated model, with minimized spillage, operates more frequently on the highest relative efficiency than the initial one, even though the prices are the same. This can be seen in Fig.5 and Fig.7. When setting the spillage to zero by having a higher fixed end reservoir content, the model recognizes that it has to save water, in other words not spill, to reach the end values. Reducing the initial high amounts of spillage is also more applicable to reality.

As a tool for modeling energy systems, Spine comes with both positive and negative qualities. First of all we experienced the documentation for the software lacking some information and quality. During our work we also encountered some technical issues and bugs with the software, which slowed down our work. However, Spine brings along a lot of opportunities for building and modeling a complex energy system. It is systematic and flexible as well as educational since it makes it easy to visualize and understand the systems. As of now, Spine is not very user-friendly but has a great potential for being developed into a very useful tool for modeling and optimizing energy systems.

V. CONCLUSION

In this project a model of Skellefte River was constructed and optimized with respect to electricity prices, by using the software Spine. A piece-wise linear dependency on the electricity production was successfully applied to increase the optimization accuracy. The work done in this project will also contribute to updates of the online documentation of the software.

For future studies it would be interesting to develop our model further, for example by assigning a value for stored water instead of having a fixed end reservoir level. It is interesting to investigate the regulating capacity of hydropower when there are extreme changes in energy demand. This has been done previously in Spine but with linear dependencies. Optimizing with a piece-wise linear dependency would give more accurate results.

APPENDIX

ELECTRICITY PRODUCTION ON THE HYDROPOWER PLANTS

ACKNOWLEDGMENT

We wish to thank our supervisor Mikael Amelin for his professional guidance throughout the whole project. We would also like to thank Iasonas Kouveliotis-Lysikatos for technical support with Spine.

REFERENCES

- [1] Our world in data. (2022, May) Emission by sector. England. [Online]. Available: <https://ourworldindata.org/emissions-by-sector>
- [2] IPCC, "Climate Change 2022: Impacts, Adaptation, and Vulnerability. Contribution of Working Group II to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change," *Cambridge University Press*, Feb. 2022.
- [3] K. S. V. Santhanam, *Introduction to hydrogen technology*, 2nd ed. New Jersey: Wiley, 2018.
- [4] LKAB. (2021, Sep.) Hybrit – för fossilfritt stål. LKAB, Luleå, Sweden. [Online]. Available: <https://www.lkab.com/sv/om-lkab/teknik-och-processutveckling/forskningssamarbeten/hybrit--for-fossilfritt-stal/>
- [5] IEA. (2022, May) Trends and developments in electric vehicle markets. LKAB, Luleå, Sweden. [Online]. Available: <https://www.iea.org/reports/global-ev-outlook-2021/trends-and-developments-in-electric-vehicle-markets>
- [6] European Commission. (2022, Mar.) Repowereu: Joint european action for more affordable, secure and sustainable energy. European Commission, Strasbourg, France. [Online]. Available: https://ec.europa.eu/commission/presscorner/detail/en/ip_22_1511
- [7] G. Sidén, *Förnybar energi*, 2nd ed. Lund, Sweden: studentlitteratur, 2015.
- [8] (2021, Feb.) Case study a5 tutorial. [Online]. Available: https://spine-toolbox.readthedocs.io/en/latest/case_study_a5.html
- [9] F. Lien-Oscarsson and T. Sibo, "Modellering av vattenkraft i spine-en studie om gränserna för vattenkraften som reglerande energikälla i framtidens elsystem," KTH, Stockholm, Sweden, Bsc. Thesis, 2021.
- [10] P. Breeze, *Hydropower*, 1st ed. London, United Kingdom: Elsevier Science, 2018.
- [11] P. Hogselius, *När folkhemselen blev internationell: elavregleringen i historiskt perspektiv*, 1st ed. Stockholm, Sweden: SNS förlag, 2007.
- [12] R. Östberg. (2020, Feb.) Vattenkraft. [Online]. Available: <https://www.energimyndigheten.se/fornbyart/vattenkraft/>
- [13] G. Sidén, *Förnybar energi*, 1st ed. Lund, Sweden: Studentlitteratur, 2009.
- [14] L. Söder and M. Amelin, *Effektiv drift och planering av kraftsystem*, 11st ed. Stockholm, Sweden: KTH, 2011.
- [15] A. Waernlund, "Modelling the nordic hydro power system with spine toolbox," Master's thesis, KTH, Stockholm, Sweden, 2021.
- [16] M. Sunnefors and T. Vainionpää, "Optimering av ett småskalig vattenkraftssystem," Master's thesis, KTH, Stockholm, Sweden, 2005.

- [17] X. lin Ge, L. zi Zhang, J. Shu, and N. fan Xu, "Short-term hydropower optimal scheduling considering the optimization of water time delay," *Electric Power Systems Research*, vol. 110, p. 188, 2 2014.
- [18] B.Sandström, "Cost-effective capacity expansion of hydropower plants in skellefteälven," Master's thesis, KTH, Stockholm, Sweden, 2019.
- [19] Skellefteå kraft. (2022, Apr.) Välkommen till finnfors kraftverksmuseum. Skellefteå kraft, Skellefteå, Sweden. [Online]. Available: <https://www.skekraft.se/om-oss/foretaget/finnfors/>
- [20] C.Flood, "Hydropower in sweden: An investigation of the implications of adding detail to the modelling of hydropower in osemosys," Master's thesis, KTH, Stockholm, Sweden, 2015.
- [21] Skellefteåälvens Vattenregleringsföretag. (2022, Apr.) Skellefteåälvens vattenregleringsföretag svf. Skellefteåälvens Vattenregleringsföretag, Skellefteå, Sweden. [Online]. Available: <http://www.skelleftealven.se/>
- [22] (2022, Mar.) Spine. [Online]. Available: <http://www.spine-model.org/>
- [23] I. K. Lysikatos, M. Marin, M. Amelin, and L. Söder, "Exploring multitemporal hydro power models of the nordic power system using spine toolbox," *Institute of Electrical and Electronics Engineers*, 2020.

CONTEXT H

DESIGN AND TESTING OF NOVEL MICROWAVE/ ANTENNA TECHNOLOGIES

POPULAR DESCRIPTION

Microwaves are not only used for heating your lunch

Whenever you send a text message, stream a movie, or call someone on a wireless device, it's very likely that the information is being transmitted using microwaves. There are many areas in microwave technology that are currently being researched, such as new exciting forms of communication and even ways to treat cancer. Thinking this sounds spectacular? That is the power of microwave and antenna technology.

Electromagnetic waves are all around us, even if we are not aware of it. Our bodies emit in the infrared frequencies and can also sense frequencies in the visible region of the spectrum, more commonly known as light. In that sense, microwaves can be explained as invisible light at lower frequencies. Like light, these can be transmitted wirelessly. This is done using antennas that can send and receive microwaves. Antennas exist everywhere. For example, they are used in your mobile phone and by medical professionals.

Antennas are not only a human invention. Many insects, such as bees, use their antenna to perceive information of their surroundings and to be interconnected within their colonies. Wireless communication, such as with 5G, could be seen as a honeycomb in which all of our high-tech devices are connected with each other.

The use of microwave technologies can be extended to medical applications. For example, researchers at KTH Royal institute of technology study how microwaves could be used in radiotherapy, as a new and better cancer treatment method. This field of research is constantly growing, suggesting that microwaves will be of great importance in future medical treatments.

Microwave and antenna technologies are relevant to our daily lives. They improve our societies through enabling us to communicate with each other, treat cancer and allow bees to find their way home. This shows how microwaves and antennas have a major impact on our society.

SUMMARY OF PROJECT RESULTS

The technologies of today and tomorrow are increasingly using wireless communication. In order to communicate wirelessly, an antenna has to be used. As lower frequencies are crowded, using higher frequencies is needed. At these frequencies, new challenges arise. Therefore, the need for developments within microwave technologies has never been greater. The project groups in context H focus on electromagnetic fields and wave propagation. These phenomena are studied through applications relying on microwave technology. Applications include communication, medical technology, defense, remote sensing and radar technology.

Project group H1 has designed and manufactured a 3D printed geodesic lens antenna. This antenna type consists of a curved parallel plate that is rotationally symmetric. When electromagnetic waves pass through this structure it acts as a lens. The curvature is adjusted to achieve specific behaviors. The rotational symmetry enables beams in different directions by using

multiple feed ports along the perimeter of the lens. The curvature of the lens is based on a defocused Luneburg lens which achieves good coverage between beams at discrete angles. The lens has been folded in on itself to achieve a smaller profile. Geodesic lens antennas have a lot of potential in future applications such as 5G and space communications.

Project group H2 has investigated how different periodic structures on a printed circuit board affect the board's filter properties. The investigated properties were the unwanted signal rejection strength and the amount of frequencies that were considered unwanted. A stop band is the range of frequencies that are considered unwanted and the signal rejection strength is the rate at which signals in this range are weakened. Having a filter with a periodic pattern means that it can easily be printed on a circuit board. Filters with this type of design can then be attached to a receiving antenna in order to eliminate signals with unwanted frequencies. The goal was a broad stop band and a strong signal rejection. The filter design was done using two different methods; a commercial simulation software and a novel method that can yield the rejection strength. Two filters with different geometries were designed and compared with each method. The two filters were manufactured and the empirical results confirmed the simulated results. The novel method to yield and compare the filter's rejection strength could potentially be used for every filter. This can potentially lead to stronger signal rejection strengths when designing periodically repeating filter structures.

Project group H5 has focused on the design and manufacturing of an antenna prototype with intended use for imaging applications, mainly radar. This antenna is able to change the beam direction with the frequency at which the antenna is fed, which is useful to scan over different regions in space. This can be achieved with other methods at the cost of introducing errors in the aiming capabilities and they require a bigger size of the whole system. One of the main issues of frequency scanning is radiation over the perpendicular direction, as due to physical reasons, the radiation efficiency degrades and reflections are introduced. To overcome this, a design with a so-called glide symmetric Goubau line has been proposed to suppress unwanted reflections. Another important feature for imaging applications is to reduce the frequency range at which the antenna operates. The frequency bandwidth required to scan from forward to backward directions can be reduced by introducing radiation patches enabling the antenna to scan faster. The performance of the designed prototype has been analyzed using simulation software. Future works could focus on the enhancement of the radiation efficiency and inclusion of simultaneous operation at different frequencies.

Project group H6 studied a possible cancer treatment method that is using gold nanoparticles under microwave radiation to achieve local heating of tumors. All materials, including human tissues, have electromagnetic properties and will therefore be affected by electromagnetic waves. To study how human tissue behaves under microwave radiation, a frequency dependent electromagnetic model of tissues containing tumors was created. Analysis of the modeled tissues were done by comparing exact analytical equations and numerical simulations of the energy absorbed in the material. The project group has shown correspondence between analytical and simulated results using the tissue models created, with results that suggest a possible future use of this cancer treatment method. Future work could focus on optimization of the tissue models or a study about other types of electromagnetic waves.

Microwave technologies may be the basis of many future technologies relevant to our daily lives. In order to obtain technological achievements such as faster communication, more advanced cancer treatment and better satellites, microwave engineers have to design and implement devices using frequencies in the microwave range. Using higher frequencies comes with a plethora of challenges, such as increased energy consumption due to increased attenuation. All the subjects presented in this context present novel theory and implementations within these research areas, and can be developed further to benefit society.

IMPACT ON SOCIETY AND ENVIRONMENT

Advancements in microwave technologies impact individuals, our societies and the environment. The impacts of different applications of microwave technologies are discussed. Applications include communications, defense and health care.

Specific ethical problems related to developments within microwave and antenna technologies are also discussed, as well as sustainability within the context of power consumption and material usage.

Electromagnetics and microwave systems have a lot of applications within warfare, both for offensive and defensive means. One of the primary uses for this technology within warfare is radar. Radar surveillance systems can give early warnings of attacks, which can help civilians evacuate and military personnel make necessary preparations and thus save a lot of lives. Radar can also be used for more offensive measures, such as missile detonation and guidance systems. More precise weapons systems can be used to reduce collateral damage, but they can also be used to cause more casualties.

With an ever increasing demand for new medical treatments of cancer patients, microwave technologies may be a leap towards moving away from today's hazardous radiation treatments of cancer. It could also be used in novel imaging techniques that could result in higher resolution biomedical images. Such images could contribute to a higher rate of cancer detection at a lower cost. These applications could lower the cost for high quality treatments and be very beneficial for cancer patients.

Microwave technologies and antennas are essential for any sort of wireless communications. New systems, such as 5G, are available to support a higher rate and more reliable data transfer. These technologies enable the use of higher frequency bands in order to accommodate more users. The design for such frequencies results in a reduction of the size of microwave devices due to the reduction in wavelength. This has a lot of advantages in terms of integration, but comes at the cost of a decrease in efficiency. Higher frequencies are also more severely attenuated and smaller devices tend to be more power hungry.

Many people do not reflect upon how much power microwave devices consume. Since there is a shortage of clean energy in the world today and our planet is subject to global warming, engineers today should have the obligation to design energy efficient devices. Furthermore, as society develops new standards for technologies, consumers are to some degree forced to purchase new devices and waste their former. Engineers should therefore always reflect on whether the benefits of new technology outweigh the consumption of power and materials. The choice of material in one's design is critical when regarding sustainable production and recycling. For instance, one of the main drawbacks of using antennas and microwave technology in space is that at the end of the mission lifetime, they end up becoming space waste. This is why new space technologies are opting for smaller satellite payloads.

Many ethical dilemmas arise when developing microwave technologies. One clear example is that the technologies can be used in military contexts. Should engineers therefore stop developing them? We believe that the engineer has to be especially cautious in these cases. If designing tools used for killing people is considered axiomatically detrimental, a deontological argument could be made for avoiding development of warfare technology altogether. Viewed through a utilitarian lens, the morality of warfare technology development depends a lot on its usage.

From a communications point of view, the world is getting interconnected in such a way that it may surpass the limits of privacy; not only regarding the handling of personal information, but also the possibility to track individuals. The ultimate dilemma is how to develop beneficial technologies without infringing upon people's personal integrity. From a sustainable viewpoint, engineers have a responsibility not to persuade customers into buying wasteful equipment and to strive for power optimized designs. This is to minimize the overall effect on the environment, which is of great importance.

Microwave technology presents a significant benefit to society and individuals within wireless communications, medical applications and warfare. However, microwave technology also contains major challenges within sustainability and ethics that need to be tackled.

3D Printed Modulated Geodesic Lens Antenna With Even Coverage in the Far-Field

Harald Lindohf and Marcus Wikner

Abstract—The development of 5G and 6G entails new demands on antennas. This includes fast and reliable connections to a large number of devices. A wider area of coverage, and thus more antennas are also expected, which is problematic for the expensive antennas used today. To meet those demands, a geodesic lens antenna has been proposed. The antenna utilises several feeding ports for beam forming. It is designed to operate at a frequency of 8 to 12 GHz and is optimised to have an even coverage in the far-field. The design is modulated with one fold to reduce the height of the antenna. A prototype of the antenna is 3D printed with PLA and coated with aluminium tape. The design has a simulated realised gain of 13.5 dBi and beam width of around 30°. The 3D printed antenna could not be tested due to technical problems with the testing facilities, but is expected to have similar results.

Sammanfattning—Med utvecklingen av 5G och 6G kommer stora krav på antenner. Flera enheter skall kunna vara uppkopplade och samtidigt krävs högre hastigheter med stabil uppkoppling. Utöver det ställs det även krav på en bred täckning vilket innebär att fler antenner behöver kopplas upp, vilka har höga kostnader idag. För att möta dessa krav har en design för en geodetisk linsantenn lagts fram. Antennen använder flera ingångar för att skapa en riktbar stråle. Den är designad för att operera inom frekvenserna 8 till 12 GHz och är optimerad för att få en jämn täckning i fjärrfältet. Designen nyttjar en vikning för att minska antennens höjd. En prototyp av antennen tillverkas med hjälp av 3D printad plast som beläggs med aluminiumtejp. Designen har en simulerad förstärkning av 13.5 dBi och en strålbredd runt 30°. Den 3D printade antennen kunde inte testas på grund av tekniska problem med testutrustningen men förväntas ha liknande resultat som den simulerade.

Index Terms—3D-printed, geodesic, modulated, Luneburg, beamforming, lens antenna

Supervisors: Sarah Clendinning, Shiyi Yang, Oscar Quevedo-Teruel

TRITA number: TRITA-EECS-EX-2022:148

I. INTRODUCTION

As society develops, the world gets more and more connected. With the development of 5G and 6G, a reliable and wide frequency spectrum needs to be supported. To increase data transfer rates, higher frequencies are introduced which are more sensitive to losses. Since a wide coverage requires many antennas, cost is also a big factor [1]. An economical flexible system is therefore of high interest, and one solution is the lens antenna.

Today, the most common beamforming solution without moving parts are array antennas. These are at high frequencies expensive to implement, and their complex feeding network causes losses. To overcome this, one solution is a rotationally symmetrical lens antenna. These antennas are cheaper to

implement and have a less complex feeding network, which makes beamforming much easier [2] [3]. An early example of a rotationally symmetric lens is the Luneburg lens, which was theorised by Rudolf Luneburg in 1944 [4]. A Luneburg lens generates a planar wave from a point source positioned on the periphery of the lens. One implementation of Luneburg lenses are dielectric lenses with a graduated refractive index. However, the permittivity of the dielectric yields losses, especially at higher frequencies. To overcome this, a geodesic design uses a curved parallel plate structure with a homogeneous refractive index, which yields lower losses at high frequencies. The waves take the locally shortest path through the lens curvature, as given by Fermat's principle. By selecting an appropriate lens curvature, one can mimic the optical path through a graded index lens, and thus achieve the same focus characteristics. In addition to Luneburg lenses, The geodesic approach may generalized to achieve different focus qualities, including that of the Maxwell fish-eye lens, where the focus point is located at the lens periphery [5].

One challenge with using multiple feed ports as a method of beamforming is that it is discontinuous. If two neighboring feed ports produce beams that are 15° apart, the antenna will have a lower gain between these beams. One way of addressing this is to design the antenna in such a way that it produces wider beams, which gives a more even far-field gain [6]. For this paper, a design for a 3D printed geodesic lens antenna with improved crossover levels is proposed and tested. Software used is CST Studio Suite, MATLAB and Fusion 360.

II. THEORY

A. Antenna far-field patterns

The far zone of an antenna is the region where the distance R to the antenna is much larger than its operating wavelength. The radiated fields in this region are called the far-fields. The relative intensity of the far-field is often plotted as a function of polar coordinates ϕ and θ . Such a plot is called the far-field pattern of the antenna, and illustrates in what directions the antenna radiates. Since the far-field intensity U depends on the input power it is pertinent to normalize it against another parameter. Given a total radiated power P_r . The directive gain G_D is defined as

$$G_D = \frac{U}{P_{av}} = \frac{4\pi \cdot U}{P_r} \quad (1)$$

where $P_{av} = \frac{P_r}{4\pi}$, which is the average radiation intensity [7].

Another alternative is the realised gain, which is defined as

$$G_{\text{realised}} = \frac{U}{P_{\text{in}}} \quad (2)$$

where P_{in} is the input power used to excite the antenna [7] [8].

Beam width is a scalar parameter that describes how sharp the main beam is. It is commonly defined as the angle width between the points at which the far-field gain is half of its maximum value (-3 dB). In this paper, 2dB beam width is also considered, which is defined as the angle between the -2dB points.

Antennas will typically radiate in several unwanted directions, which causes sidelobes in the far-field pattern. This is an important parameter that is called sidelobe level (SLL). The SLL is defined as the gain of the first sidelobe compared to the main beam.

B. Geodesic lens antennas

Lens antennas use a lens in order to manipulate electromagnetic waves and achieve specific beam qualities. One type of lens used in antennas is a geodesic lens, in which electromagnetic waves are confined between two curved parallel plates. Fermat's principle shows that the waves will take the locally shortest path through the curvature, and are thus focused. This happens without any changes in refractive index. Examples of this can be seen in Fig. 1. In both lenses, a point source is used to send waves across a curved surface. In Fig. 1A, the waves travel across a spherical surface, which causes them to be focused into another point on the other side of the lens. This is an example of a Maxwell fish-eye lens. In Fig. 1B the curvature is different, and the waves are focused into parallel waves. This is an example of a Luneburg lens. Their respective profiles are shown in Fig. 1C.

An arbitrary focal distance can be achieved by designing a specific lens curvature. Since the lens is rotationally symmetric, the curvature is defined by the lens profile $Z(\rho)$, where Z is the height of the lens at a distance ρ from its center. Consider a lens of radius 1, with a point source at a radius r_1 and a desired focus point at a radius r_2 . The length $S(\rho)$ from the center of the lens to a radius ρ along the profile curve can be written as

$$S(\rho) = A\rho + B \arcsin \rho \quad (3)$$

where A and B are given by

$$A = 1 - \frac{1}{\pi} \arcsin \sqrt{\frac{1-\rho^2}{r_1^2-\rho^2}} - \frac{1}{\pi} \arcsin \sqrt{\frac{1-\rho^2}{r_2^2-\rho^2}}, \quad (4)$$

$$B = (M-1) + \frac{1}{\pi} \arcsin \frac{1}{r_1} + \frac{1}{\pi} \arcsin \frac{1}{r_2}. \quad (5)$$

$M \cdot \pi$ is the angle between the point source and the focus point. The definitions of M , r_1 and r_2 are illustrated in Fig. 2. In Fig. 1 $r_1 = 1$ and $M = 1$. For the Maxwell fish-eye lens $r_2 = 1$ and for the Luneburg lens $r_2 = \infty$ [5].

This type of lens can be implemented using a curved parallel plate waveguide. The lens profile is calculated according to

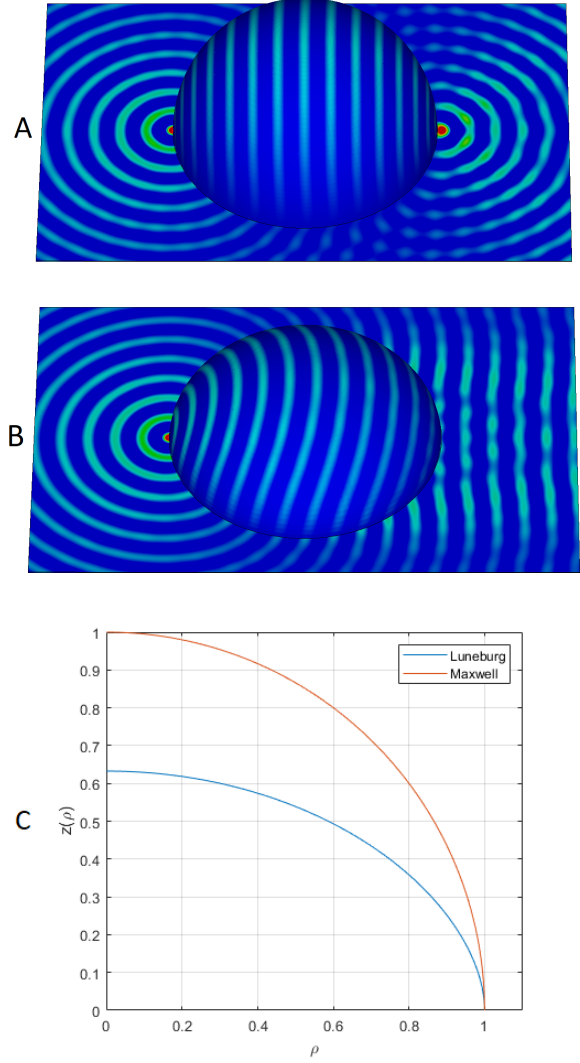


Fig. 1. Simulated Maxwell and Luneburg lenses, A and B respectively. A point source on the left side of the perimeter is used to excite both lenses. Figure C illustrates the profiles of the different lenses.

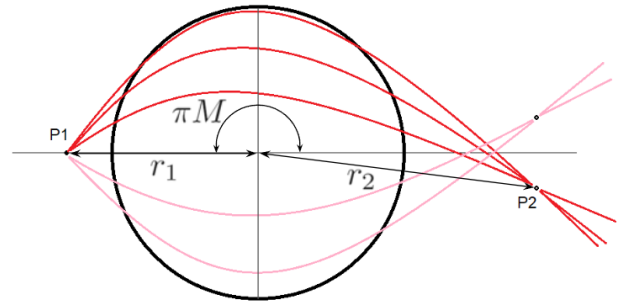


Fig. 2. Geodesic lens antenna with light rays shown in red and pink.

equation (3), and fits in the center between two parallel plates. Waves that travel between the plates will be confined to the calculated lens curvature. Since the geodesic lens is rotationally symmetric, it can be excited from any angle. Multiple feed ports along the perimeter of the lens can thus

be used to create a beam in multiple directions. This type of beamforming is one of the major advantages of geodesic lens antennas. If each feed port has a fixed angle, beamforming will only be possible in fixed directions. The far-field gain in between two such directions, where two beams meet, will be lower compared to the max gain of each beam. This ratio is called the crossover gain. In order to achieve lower crossover gain, and thus more even coverage in the far-field, one can select foci points that yield a wider beam width.

C. Scattering parameters

When characterising a microwave system it is often useful to view it as a network with one or more ports. The scattering parameters (S-parameters) of any n port network is defined as

$$\begin{bmatrix} V_1^- \\ V_2^- \\ \vdots \\ V_n^- \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1n} \\ S_{21} & S_{22} & \cdots & S_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ S_{n1} & S_{n2} & \cdots & S_{nn} \end{bmatrix} \begin{bmatrix} V_1^+ \\ V_2^+ \\ \vdots \\ V_n^+ \end{bmatrix} \quad (6)$$

where for every port i , V_i^+ is the amplitude of the voltage wave that travels from the port into the network, and V_i^- is the amplitude of the voltage wave that travels from the network into the port. If only port j is excited, S_{ij} describes the portion of the excitation wave that makes it into port i . Accordingly, S_{ii} describes the reflection seen by port i . The S-parameters can be directly measured using a vector network analyser (VNA) [8].

D. Waveguide

A waveguide is a type of transmission line that consists of a metallic pipe through which electromagnetic waves may propagate. There are two types of modes that can propagate through a waveguide, These are transverse electric (TE) and transverse magnetic (TM) modes. The electric field of a TE wave is transversal to the propagation direction, whereas the magnetic field has a parallel component. The same logic applies for TM waves. For a rectangular waveguide with side lengths a and b , where $a \geq b$, the cutoff frequency of a given mode can be calculated as

$$(f_c)_{mn} = \frac{1}{2\sqrt{\mu\epsilon}} \sqrt{\left(\frac{m}{a}\right)^2 + \left(\frac{n}{b}\right)^2} \quad (7)$$

where m and n define the mode in question. For TM modes, m and n have to be nonzero. For TE modes, m or n can be zero. Hence, the TE₁₀ has the lowest cutoff frequency, and is therefore called the fundamental mode. A mode can only propagate if the frequency is below f_c [7]. Fig. 3 illustrates the propagation of waveguide modes TE₁₀ and TE₂₀.

Another type of transmission line relevant to this work is the parallel plate waveguide (PPW). It consists of two parallel conductive plates, between which electromagnetic waves may propagate. Besides TE and TM modes, the PPW can also propagate TEM modes, in which both the electric and magnetic fields are transversal to the direction of propagation.

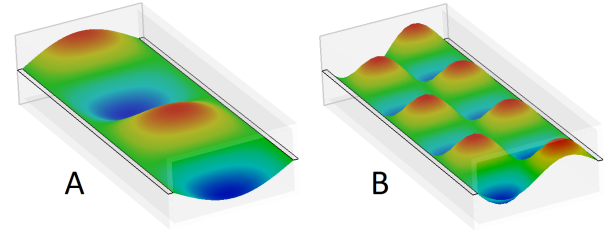


Fig. 3. Simulated propagation of TE₁₀ mode (A) and TE₂₀ mode (B) through a rectangular waveguide.

The TEM mode of a PPW has no cutoff frequency, but TE and TM modes do. Their cutoff frequency can be calculated to

$$(f_c)_n = \frac{n}{2d\sqrt{\mu\epsilon}} \quad (8)$$

for both TE_n and TM_n modes. Where d is the plate separation [8].

III. DESIGN

A. Lens Profile

The lens is designed for operation between 8 and 12 GHz. It consists of a curved PPW. A plate separation of 6 mm is chosen so that only the fundamental TEM mode propagates below 12 GHz. One design goal is for the entire antenna to be no larger than 300×300×60 mm (W×D×H). In order to achieve this, a lens radius of 75 mm was chosen. This corresponds to 2.5λ at 10 GHz.

In order to calculate the lens profile, a MATLAB script was written. From given parameters r_1 , r_2 and M , the script calculates $S(\rho)$ according to equation (3). The program then calculates $Z(\rho)$ by stepping through $S(\rho)$ with a constant step length δS . For each step, the program finds the corresponding increase in radius $\delta\rho$, and calculates $\delta Z = \sqrt{\delta S^2 - \delta\rho^2}$, which is added to the profile height. The script also calculates profile functions for the top and bottom plate of the PPW.

In order to reduce the overall height of the antenna, the lens can be modulated. By folding the lens profile in on itself the height is reduced while maintaining the profile length $S(\rho)$, and thereby the focusing characteristics of the lens [9]. An example of this is illustrated in Fig. 4. For a profile with n folds, the folded height can be derived as

$$h_f = \frac{h_0 + \delta}{n + 1/2} \quad (9)$$

where h_0 is the height of the unfolded profile and δ is the vertical distance from the bottom of the unfolded lens to the center-line of the folded lens. This is consistent with [9]. The MATLAB script that calculates the profile curve also applies the modulation.

Due to the small lens radius compared to the plate separation, only one fold is applied to the lens. the distance δ is set to 3 mm. The flare and feed port were modeled flush with the bottom of the lens, which adds a 3 mm vertical bias to these components. Setting δ to 3 mm yields a common center-line for the feed port, flare and the folded lens profile.

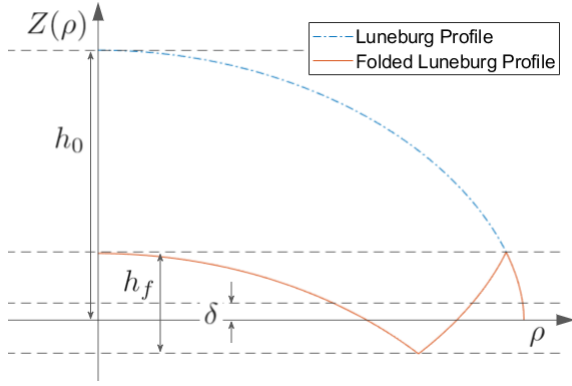


Fig. 4. A Luneburg lens profile before and after applying two folds.

An important design goal was for the realized far-field gain where two beams overlap to be within 2 dB of the maximum gain, given that the two beams are 15° apart. This can be ensured by having a 2 dB beam width above 15° . To achieve this, various values of r_2 and M were simulated with r_1 fixed at 1.03. The addition of the modulation has a significant impact on the simulated far-field patterns. Therefore, optimisation of the far-fields has to be redone after applying the folds. The selected final values are $r_2 = 1.4$ and $M = 1.02$, which yields a 2 dB beam width of 29.1° at 10 GHz for a single port. Fig. 5 shows the final lens profile.

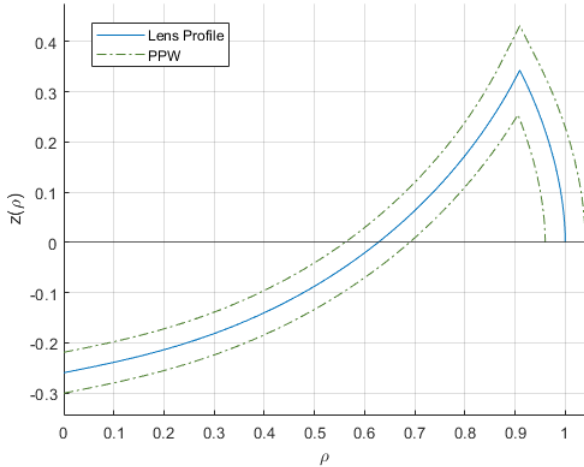


Fig. 5. Final lens profile with PPW profiles shown.

B. Chamfer design

Chamfers have been introduced in order to reduce the reflections caused by sharp turns in the lens, see Fig. 7. The first chamfer is located near the perimeter of the lens. The second chamfer is located where the lens is folded. The chamfer that is close to the perimeter was designed before applying the modulation. In order to find the optimal chamfer geometry, a parameter sweep was done. To reduce the time required for simulations, the sweep was performed on a simplified PPW model of the lens profile, shown in Fig. 6. The results are

then confirmed with a finer sweep on the complete model. The same process was repeated after applying the modulation of the lens. The parameters that define the chamfer geometry are shown in Fig. 7. The resulting parameters are listed in Table I.

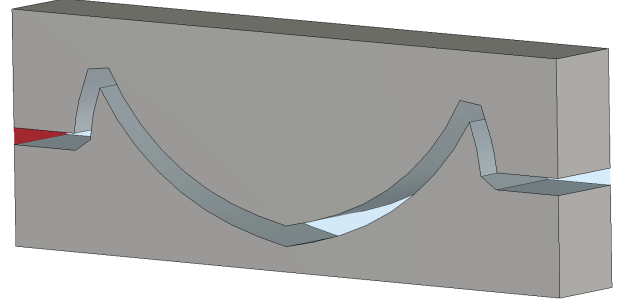


Fig. 6. PPW model on which chamfer optimisation is performed.

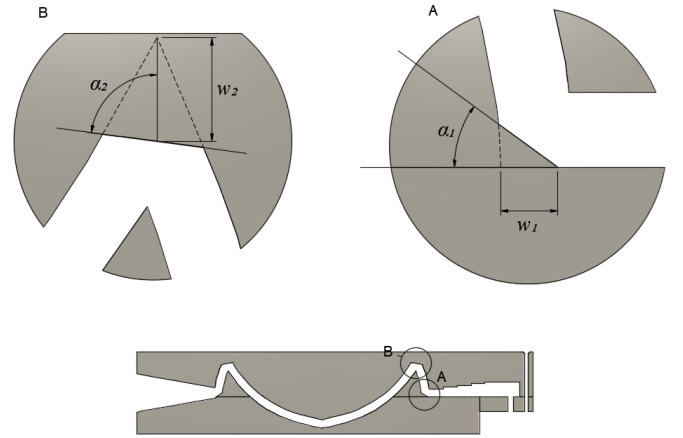


Fig. 7. Cutting plane view of the complete lens with a closer view at the two chamfers.

TABLE I

Chamfer dimensions and angles

w_1	α_1	w_2	α_2
4.6 mm	36.1°	6 mm	82.5°

C. Flare design

The primary purpose of a flare is to reduce reflections. It can also improve the directivity of the antenna. The flare is located on the lens perimeter, with its center positioned opposite of the middle feed port. It provides a transition between the PPW and open space. In order to avoid long simulation times, a PPW model is used, shown in Fig. 8. A linear flare design was selected for ease of construction. A parameter sweep was done to find the optimal dimensions. The upper sweep limit is set so that the overall size of the antenna does not exceed $300 \times 300 \times 60$ mm ($W \times D \times H$), as specified in the design goals. The selected parameters yield the best S-parameter results while also having shorter length than parameters of similar quality. The flare has a length of $l = 59$ mm and a height of $h = 27$ mm. It covers 230° of the lens perimeter.

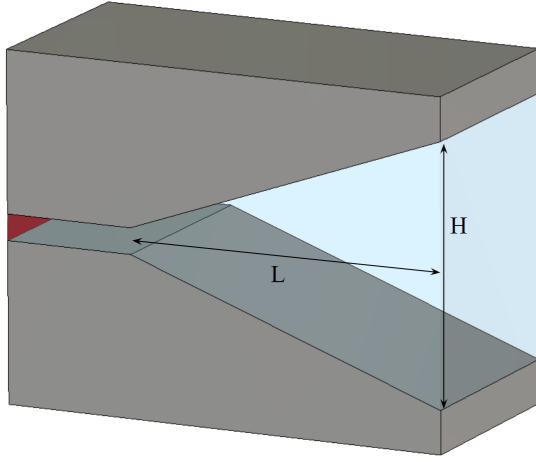


Fig. 8. PPW model used for flare optimisation.

D. Feed port design

A feed port on the lens perimeter is used to excite the antenna. The feed port, illustrated in Fig. 9, consists of a stepped waveguide. To excite the feed port and the antenna an RND 205-00498 coaxial connector is inserted in the waveguide. The 4 steps between the coaxial connector and the lens are all equal in height and spacing. The waveguide segment closest to the lens has the same height as the PPW. In order to make manufacturing easier, only one side of the feed port is stepped. Simulations demonstrate that this yields sufficient results, as shown in Fig. 10. A parameter sweep on a simplified model is used to find the optimal height, width and length of the feed port. The simplified model used does

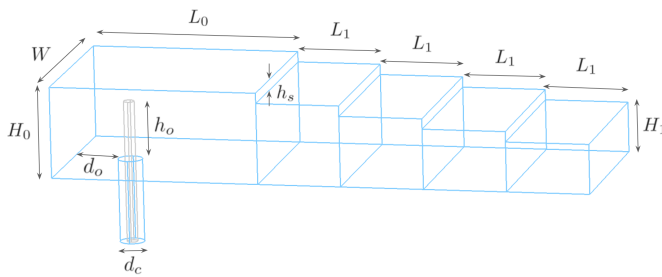


Fig. 9. Final feed port design with dimensions.

not include the coaxial connector or the PPW. The limits of this sweep were set so that only the fundamental TE_{10} mode is excited. A second sweep on another model was then performed to find the optimal insertion length and location of the coaxial connector. This model, illustrated in Fig. 11, includes the coaxial connector and a short section of PPW. A finer sweep was finally performed on the full model to refine the insertion length and location of the coaxial connector. The final parameters, listed in Table II, were selected for good S-parameter results.

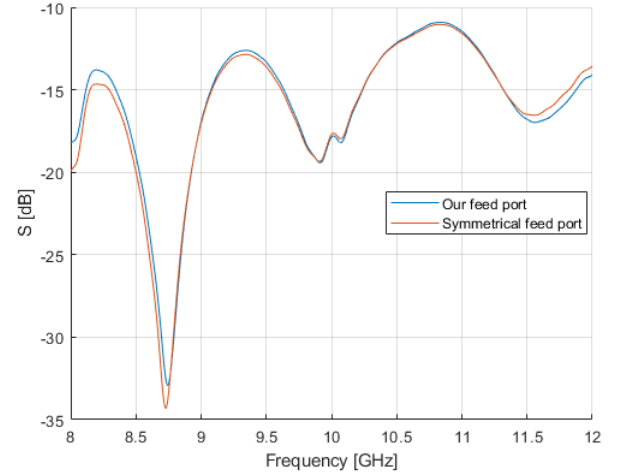


Fig. 10. S-parameters comparing our feed port and a symmetrically stepped feed port.

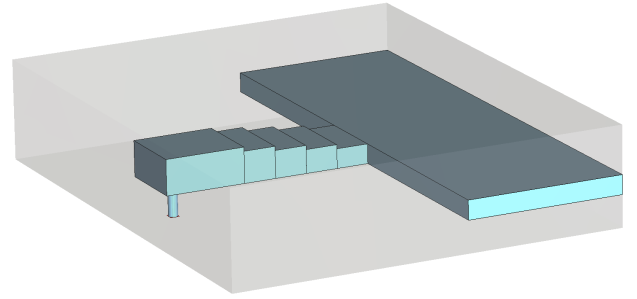


Fig. 11. Simplified simulation model with a coaxial connector and discrete port.

TABLE II

The resulting parameters of the feed port

Parameter	L_0	L_1	H_0	H_1	W
length (mm)	25	10	11	6	22
Parameter	h_0	h_s	d_0	d_c	
length (mm)	6.3	1.25	6.6	1.487	

IV. RESULTS

The final model is simulated in CST Studio Suite before beginning manufacturing. The results are shown in Fig. 12, Fig. 13 and Fig. 14. The anechoic chamber at KTH had technical problems which prevented the prototype from being tested.

A. E-fields

Fig. 12 shows E-field simulation results at 10 GHz. Each port was individually excited, resulting in a focused beam at the opposite end of the lens, which is expected. One can observe the wavefront changing shape as it passes through, and is focused by the lens.

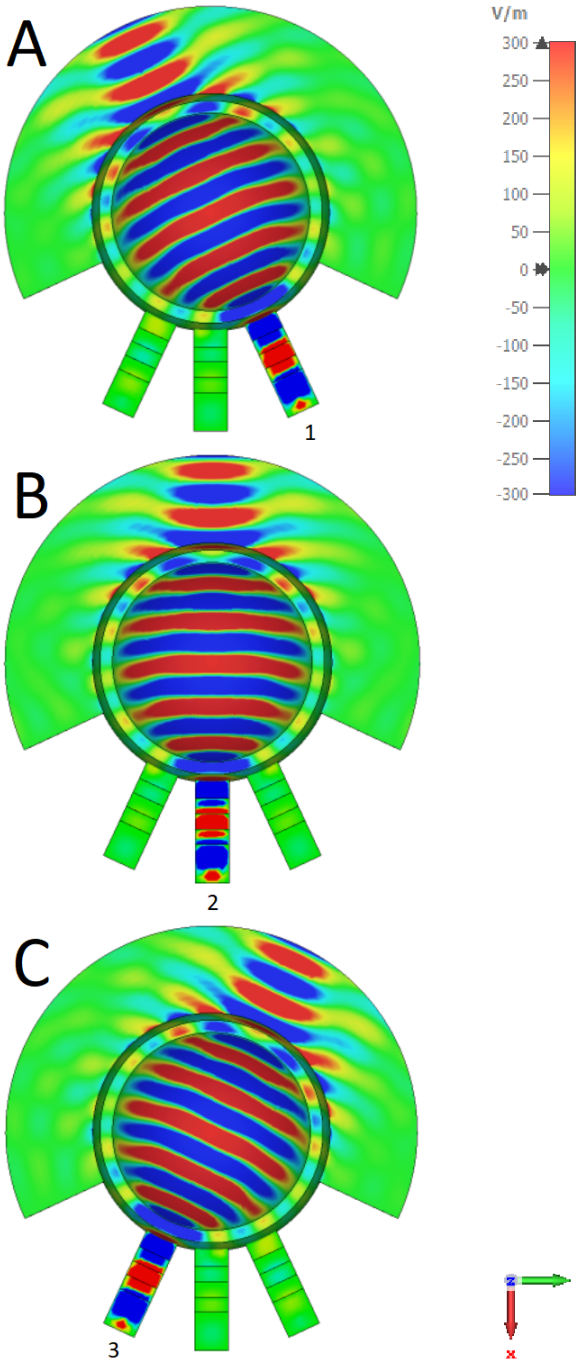


Fig. 12. Simulated E-field Z-component at 10 GHz from (A) port 1, (B) port 2 and (C) port 3.

B. Far-field

Fig. 13 shows the far-field from each port at 10 GHz. The main lobes are 25° offset from each other, which is expected since the feed ports have the same offset. The beam width is, however, lower than expected. This is caused by the antenna having multiple feed ports. Additional simulations show that the antenna achieves the expected beam width when only one port is modeled. The crossover gain with 3 ports is still below 3 dB, and an even coverage over 75° is achieved. Some asymmetries can be observed, which is unexpected as the

antenna has a symmetrical design. This has been determined to be due to numerical errors in CST.

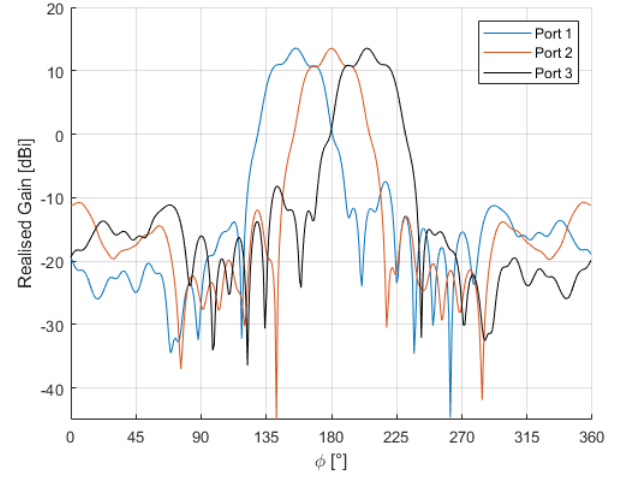


Fig. 13. Far-fields of the lens from port one, two and three at 10 GHz.

C. S-parameters

In Fig. 14 and 15 the S-parameters of the antenna are demonstrated. Due to the rotational symmetry of the lens, each port has very similar S-parameters. S_{11} , S_{22} and S_{33} are shown in Fig. 14. These values are equivalent to the total reflection seen by each port, therefore low values are desirable. For most frequencies these values are below -10 dB, which indicates acceptable matching. Fig. 15 shows S_{12} and S_{13} , which is equivalent to the amount of energy that is transmitted between the ports. Low values are desirable here as well. All values are below -20 dB, which indicates very low interference between the ports.

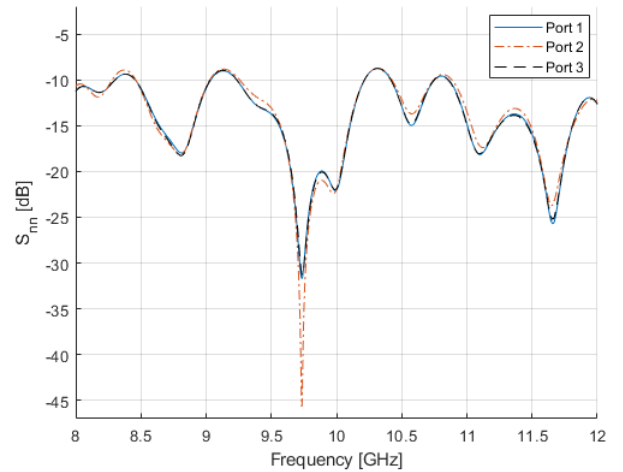


Fig. 14. S-parameters of the three different ports in 8-12 GHz.

V. MANUFACTURING

The 3D model used in simulations was imported to Fusion 360, where it was used to create a final model for manufacture.

The final model was split into 3 parts, and screw holes were added for assembly. The final 3D model is shown in Fig. 16. The model was 3D printed in PLA plastic. In order to implement the conductive components of the antenna, the 3D printed parts were covered with aluminium tape. Finally, the antenna was assembled using machine screws and nuts. A RND 205-00498 coaxial connector was attached to the feed ports. The fully assembled antenna is pictured in Fig. 17.

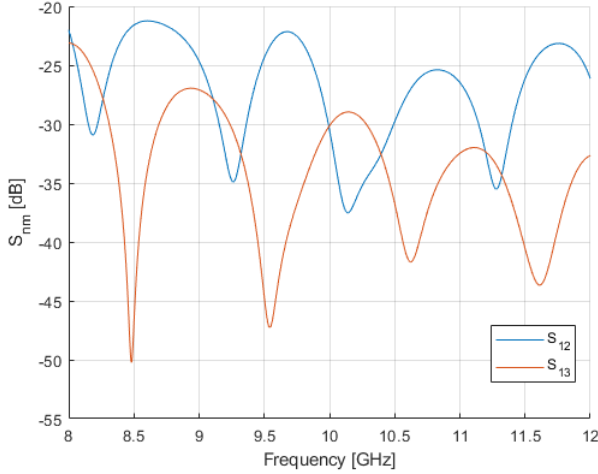


Fig. 15. S-parameters illustrating energy transfer from port 2 and port 3 to port 1.

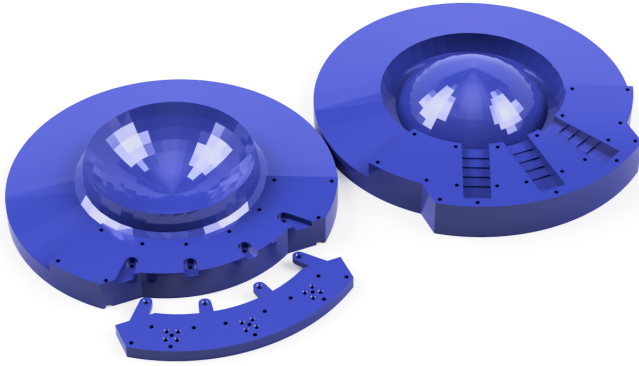


Fig. 16. The 3D model made in fusion 360.

VI. FUTURE WORK

In order to find the optimal parameters for M and r_2 (3) several simulations were made. Regression analysis was investigated as a method of predicting simulation results. Preliminary investigation showed promise, but due to time constraints a fully working model was not achieved. Investigating this further would possibly yield a time saving method of finding the previously mentioned parameters. Investigating the effect that folding, and specifically fold locations, has on far-fields is also of interest for future study.

VII. CONCLUSION

In this paper, the design (and construction) of a 3D printed geodesic lens antenna is presented. The antenna is designed

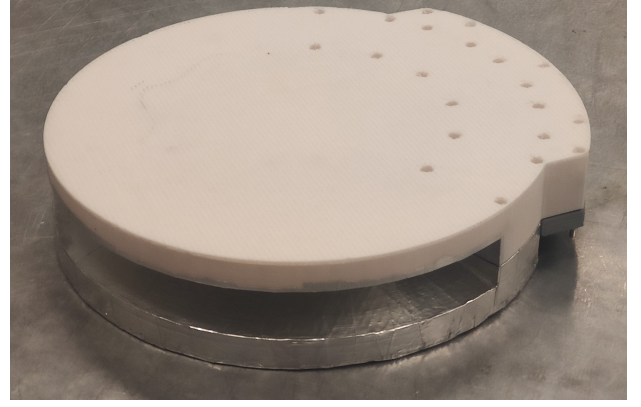


Fig. 17. The 3D printed antenna assembled.

to operate between 8 – 12 GHz. Its symmetrical design allows multiple beams to be transmitted simultaneously in different directions with low scanning losses. A scanning range of 75° with less than 3 dB of overlap between beams was achieved. A prototype was constructed using 3D printing. With a relatively simple design it is a feasible and exciting solution for future antennas.

ACKNOWLEDGMENT

The authors would like to thank Sarah Clendinning and Shiyi Yang for their invaluable guidance during this project. We would also like to thank Freysteinn Vidar Vidarsson for helping us get access to the school's computers for simulations and Pilar Castillo Tapia for helping us with 3D printing our antenna.

REFERENCES

- [1] J. Butler, in *5G Radio Technology Seminar: Exploring Technical Challenges in the Emerging 5G Ecosystem*, London, UK.
- [2] O. Quevedo-Teruel, M. Ebrahimpouri, and F. Ghasemifard, "Lens antennas for 5g communications systems," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 36–41, 2018.
- [3] P. Castillo-Tapia, Q. Liao, N. J. G. Fonseca, and O. Quevedo-Teruel, "Modulated geodesic lens antenna array," in *2021 15th European Conference on Antennas and Propagation (EuCAP)*, 2021, pp. 1–4.
- [4] R. K. Luneburg, *Mathematical Theory of Optics*. Providence, RI: Brown University, 1944.
- [5] M. Sarbort and T. Tyc, "Spherical media and geodesic lenses in geometrical optics," *Journal of Optics*, vol. 14, 07 2012.
- [6] O. Orgeira, G. León, N. J. G. Fonseca, P. Mongelos, and O. Quevedo-Teruel, "Near-field focusing multibeam geodesic lens antenna for stable aggregate gain in far-field," *IEEE Transactions on Antennas and Propagation*, pp. 1–1, 2022.
- [7] D. K. Cheng, *Field and Wave Electromagnetics*, 2nd ed. Harlow, United Kingdom: Pearson Education, 2014.
- [8] D. M. Pozar, *Microwave Engineering*, 4th ed. Hoboken, NJ: John Wiley and Sons, Inc., 2011.
- [9] J. Mitjans, "Geodesic lens antenna for sub-thz imaging," Ph.D. dissertation, KTH, Stockholm, Sweden, 2021.

A Comparison Between Applied Square and Ring CSRRs on SIWs Using the HOM Method

Carl Nordengren and Johan Bellbrant

Abstract—The rise of connected devices and the internet of things has increased the need for systems capable of transmitting high frequency signals wirelessly. An important part of these systems are the filters. Filters remove signals within unwanted frequency ranges. These filters can be implemented using e.g. periodic structures. In this article, we present a design for such a filter that aims to have a stopband between 3-6 [GHz] using square complementary split ring resonators (CSRR) on a substrate integrated waveguide (SIW). The design is based on a dimensional parametric study. An alternative design based on circular CSRR's is also presented and discussed. The design is validated using a commercially available software and a novel method simulating higher order of modes (HOM). The novel simulation method is shown to be advantageous due to its ability to evaluate the attenuation coefficient of a periodic filter. Additionally, a quadratic CSRR structure was shown to have a larger stopband and a similar attenuation coefficient when compared to circular CSRR structure when applied on a SIW. Furthermore, an impedance matching structure for the both CSRR filters were designed and both filters were simulated.

Sammanfattning—Förekomsten av uppkopplade enheter och användandet av sakernas internet har ökat behovet av system som kan sända högfrekventa signaler trådlöst. En viktig del av dessa system är filter, som eliminerar signaler inom oönskade frekvensband. Dessa filter kan implementeras med periodiska strukturer. I denna rapport presenterar vi en design för ett sådant filter med ett stoppband mellan 3-6 [GHz] som använder sig av kvadratiske "complementary split ring resonators" (CSRR) på en "substrate integrated waveguide" (SIW). Designen är baserad på en geometrisk parametrisk studie. En alternativ design som använder sig av cirkulära CSSRs presenteras och diskuteras. Den föreslagna designen valideras med en kommersiellt tillgänglig och en egenframställd metod vid namn "higher order of modes" (HOM) metoden. Den egenframställda simulationsmetoden visas vara fördelaktig då den är kapabel att evaluera filtrets attenuationskoefficient. Utöver detta visas att en design baserad på kvadratiske CSRRs vara fördelaktig då den genererar ett större stoppband och liknande attenuationskoefficient jämfört med den cirkulära CSSR designen vid tillämpning på en SIW. Fortsättningsvis presenteras en matchande struktur för båda filter varpå båda kompletta filter simuleras.

Index Terms—Split-ring resonators, bandgap-filter, higher order modes method, substrate integrated waveguide, periodic structures.

Supervisors: Chen Mingzheng, Freysteinn Vidar Vidarsson, Oscar Quevedo Teruel

TRITA number: TRITA-EECS-EX-2022:149

I. INTRODUCTION

DUE to the increase of connected devices and demand for higher data transmission rates, filters have to be designed

in order to meet these demands [1]. Additionally, in order to properly process signals that are transmitted using microwave technology such as 5G, microwave engineers have had to develop solutions that can filter highly frequent signals in an efficient manner. This is due to the fact that conventional methods such as filtering solely by lumped components may not be suitable for signals in the microwave spectrum [2].

The study of periodic structures started in the 1960's and 1970's with the use of the Floquet theorem [3]. Filters that are based on these structures are suitable for filtering electromagnetic waves in the microwave spectrum [2]. The suitability of this kind of implementation is supported by the fact that periodic structures can be designed using higher symmetries such as glide symmetry. These have been shown to enhance critical aspects of a filter, such as the bandwidth of operation and attenuation [3].

One-dimensional periodic filters lend themselves well to be manufactured using printed circuit board (PCB) technology. This is due to their periodic nature, which simplifies the manufacturing process and decreases the overall cost [4].

Periodic filters can be designed using complementary split-ring resonators. CSRRs are planar structures that are designed using a conductive material and a substrate beneath the conductor. The structure is then realized by removing some of the conductive material in a symmetrical pattern [5], [6]. The aim of doing this is to replicate electromagnetic properties that are not found in nature. These properties, that closely resembles those of metamaterials were first studied during the 1970's [5]. Properties that were studied include the inclusion of simultaneous negative electric and magnetic permittivity [5].

SIWs are PCBs with edge viases which have an equivalent rectangular waveguide in terms of wave propagation [7]. These waveguides can be complemented with specific geometries etched on them to create filters. An example of a geometry is CSRRs etched on a SIW [3].

In this article, an electromagnetic bandgap filter is designed using CSRRs on SIWs structures by analyzing the multimode transfer matrix. The designs feature the use of glide symmetry and lends itself well to be manufactured using PCB technology. The properties that are considered are the cut-off frequencies, the width of the stopband and the rejection strength in the stopband. The novel filter provides a stopband between 3 [GHz] and 6 [GHz]. This design limitations were defined by the supervisors of this project. An alternative design using CSRRs with a different geometry is also presented. Firstly, the relevant theory is discussed. Following this are the parametric studies, simulations and results which are presented

separately for each structure. After that, a matching transition is proposed and applied to both of the structures. Conclusions are then made drawn based on the simulations of the unit-cells and full structures.

II. THEORY

A. Electromagnetic modes

The fundamental Maxwell's equations in electromagnetism together with boundary conditions give infinite solutions for how propagating electric and magnetic waves can be exited through a structure. One solution for the electromagnetic wave from these infinite solutions is called a mode. As a structure's boundary conditions are determined partially by its dimensions, the dimensions can be adjusted to change possible propagating modes. The infinite solutions for the electromagnetic waves are in an orthogonal base, which makes it possible to have several different possible propagating modes per frequency. Any type of field structure can therefore be represented using an infinite number modes with unique amplitudes and propagation coefficients. Modal analysis can be applied to discontinuities within a structure [8, p. 203]. Problems with discontinuities generate an infinite number of modes spatially close to a discontinuity. In order for modal analysis to converge with a solution given by field analysis, enough modes has to be considered.

Modal analysis is an approximative alternative to field analysis of electromagnetic fields. Fields can be broken down into transverse electric, transverse magnetic and transverse electromagnetic components in microstrip lines and waveguides. The detailed composition of modes depend heavily on the system's geometry and boundary conditions [9]. The convergence of the modal is dependent upon the number of modes considered, where additional modes result in a better approximation [8, p. 208].

B. Periodic structures

Periodic structures as defined in [10] must follow either of the following criteria:

- The structure has continuous, although periodically varying, properties.
- The structure has periodic boundary conditions.

As stated in [11], a unit-cell of a periodic structure can be associated with a transfer matrix $[T_p]$, where a unit-cell is the smallest structure possible to describe a periodic structure. Assuming that the transfer matrix of a finite structure containing N periods of unit-cells is denoted by $[T_N]$, the relation stated in (1) hold [12]. The geometrical implication of a unit-cell is illustrated in Fig. 1. The study of a periodic structure's unit-cell is therefore enough to study tendencies of a full structure's propagating waves, although with some approximating drawbacks such as non exact transfer matrices [11].

$$[T_p] = \sqrt[N]{[T_N]} \quad (1)$$

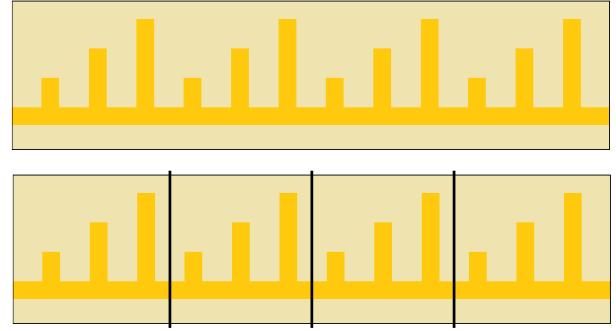


Fig. 1. A rough illustration of the relation between a complete structure (top) and its unit-cells (bottom) for an one-dimensional periodic structure. Every unit-cells transfer matrix $[T_N]$ relate with the transfer matrix $[T_p]$ of the complete structure.

C. Glide symmetry

Glide symmetry is a mathematical translation that refers to the mirroring of a certain aspect of a structure in respect to a chosen plane [13]. The plane may be arbitrarily chosen from any normal with respect to any spacial variable. Within the context of filters using one-dimensional periodic structures, the glide symmetry operator is instead defined as in (2), where the structure is periodic in the y -direction with a spatial period length of p .

$$G_z f(x, y, z) = f(x, y + p, -z) \quad (2)$$

Applying glide symmetry to a stopband filter designed using one-dimensional periodic structures has been shown to be able to increase the stopband of the filter [14].

D. Dispersion diagrams

The behavior of periodic structures can be studied using dispersion diagrams. These reveal relevant information pertaining to propagating electromagnetic modes and attenuation as a function of frequency. A dispersion diagram is a graphical representation of the propagation coefficient with respect to frequency. The propagation coefficient γ is defined in (3), where α [Np/m] is the attenuation coefficient and β [rad/m] the phase coefficient. In physical solutions to Maxwell's equations for transmission lines, both α and β can not be non-zero values simultaneously [8, p. 383]. In this article, a cut-off frequency is defined as a frequency where a transition between α and β is present. Furthermore, a stopband is defined as a frequency band where attenuation is present.

$$\gamma = \alpha + j\beta \quad (3)$$

E. 2M-port network theory

Theory pertaining to two port systems can be used when studying a filter designed using a one-dimensional periodic structure. The basic principle of two port network theory is that a system can be characterized using a transfer matrix [8, p. 188]. The elements in the matrix are frequency dependent,

and the elements can be matrices in of themselves depending on the number of modes per waveguide port [11].

A $2M$ -port system, where each port has one to M number of excited modes, the multiport (multimode) scattering matrix is defined by [11] as

$$[\mathbf{S}] = \begin{bmatrix} [\mathbf{S}_{ii}] & [\mathbf{S}_{io}] \\ [\mathbf{S}_{oi}] & [\mathbf{S}_{oo}] \end{bmatrix} \quad (4)$$

where the subscripts i and o stands for the input port and the output port respectively. Each element in (4) are sub-matrices, each having the dimensions $M \times M$.

The multiport transfer matrix in (5) can be expressed in terms of the multiport scattering matrix in (4) using the conversion in (6)-(9) for a $2M$ -port system as stated in [11]. This conversion is done for each discrete sample of frequency ω .

$$[\mathbf{T}] = \begin{bmatrix} [\mathbf{A}(\omega)] & [\mathbf{B}(\omega)] \\ [\mathbf{C}(\omega)] & [\mathbf{D}(\omega)] \end{bmatrix} \quad (5)$$

$$[\mathbf{A}] = \frac{1}{2} [([\mathbf{1}] + [\mathbf{S}_{ii}])[\mathbf{S}_{oi}]^{-1}([\mathbf{1}] - [\mathbf{S}_{oo}] + [\mathbf{S}_{io}]) \quad (6)$$

$$[\mathbf{B}] = \frac{1}{2} [([\mathbf{1}] + [\mathbf{S}_{ii}])[\mathbf{S}_{oi}]^{-1}([\mathbf{1}] + [\mathbf{S}_{oo}]) - [\mathbf{S}_{io}]] [\mathbf{Z}_o] \quad (7)$$

$$[\mathbf{C}] = \frac{[\mathbf{Z}_i]^{-1}}{2} [([\mathbf{1}] - [\mathbf{S}_{ii}])[\mathbf{S}_{oi}]^{-1}([\mathbf{1}] - [\mathbf{S}_{oo}]) - [\mathbf{S}_{io}]] \quad (8)$$

$$[\mathbf{D}] = \frac{[\mathbf{Z}_i]^{-1}}{2} [([\mathbf{1}] - [\mathbf{S}_{ii}])[\mathbf{S}_{oi}]^{-1}([\mathbf{1}] + [\mathbf{S}_{oo}]) + [\mathbf{S}_{io}]] [\mathbf{Z}_o] \quad (9)$$

Here, $[\mathbf{1}]$ is the $M \times M$ identity matrix, and $[\mathbf{Z}_i]$ and $[\mathbf{Z}_o]$ are diagonal square matrices with each non-zero element being the input impedance Z_i and the output impedance Z_o respectively. As the scattering matrix has the dimensions $2M \times 2M$, the accompanying transfer matrix will also have the dimensions $2M \times 2M$.

F. The HOM method

The HOM method presented in [11] is a method that can solve for both the attenuation coefficient and phase coefficient in a periodic structure with respect to frequency. The attenuation coefficient can not be obtained trivially using commercial simulation software. With a given transfer matrix for a unit-cell, the possible solutions to the propagation coefficient γ can be calculated as the eigenvalues to the transfer matrix as in (10) using (11). This is assuming that the transfer matrix is a good approximation of the structure. In (10), p is the period length of the unit-cell used in (2), and the elements \mathbf{V} and \mathbf{I} in (11) are $M \times 1$ array vectors of the voltages and currents respectively at the output ports. Due to the fact that the transfer matrix is frequency dependent, the frequency dependency of the propagation coefficient is implied.

$$[\mathbf{T}]\mathbf{u} = e^{\gamma p}\mathbf{u} \quad (10)$$

$$\mathbf{u} = \begin{bmatrix} \mathbf{V} \\ \mathbf{I} \end{bmatrix} \quad (11)$$

The dimensions of the transfer matrix depends on the number of excited modes. As mentioned above, an M number of excited modes implies that the dimension of the accompanied transfer matrix is $2M \times 2M$. The number of eigenvalues that are obtained from such a problem is equal to the number of rows or columns of the square matrix $[\mathbf{T}]$. This implies that the number of eigenvalues that are obtained is equal to $2M$.

The physicality of these solutions to the propagation coefficient has to be investigated when solving such a problem using the HOM-method. This can be done through a qualitative graphical analysis of the different values. This is essentially done using (3) to create a dispersion diagram that is investigated for each solution $\gamma(\omega)$. Solutions may be partially correct with respect to frequency. Multi-dimensional eigenvalue solvers solve per frequency, and may not sort continuous propagation coefficients γ correctly. This implies that a eigenvalue $\gamma(\omega)$ can be physical for a frequency band while being non-physical for another. Due to this fact, comparisons has to be made between the HOM-method and other methods to ensure that the correct solutions are obtained.

III. SIMULATIONS AND RESULTS OF UNIT-CELLS

In this project, a comparison between square and circular CSRR in SIW's has been made. Firstly, recreations of two CSRR SIW structures from [3] were made to validate the method presented in this thesis. These are 1-sided CSRR and 2-sided glide-symmetric CSRR unit-cells. Secondly, the dimensions for the 2-sided glide-symmetric structure from [3] was changed to yield the same stopband, although for a different substrate. Lastly, the same 2-sided glide-symmetric filter, although with circular CSRRs, was made with the same geometrical constraints as the former structure.

Every structure had the same simulation procedure. The phase coefficient β was obtained using CST's *Eigenmode Solver* (CST-ES), which controls if the desired cut-off frequencies are made. CST-ES does not accept open boundary conditions, which lead to the introduction of nonphysical prevalent modes which had to be manually removed. Thereafter, the multimode *Frequency Domain Solver* in CST was used, such that the multimode S-parameters for each structure were possible to obtain. Using MATLAB, the multimode S-parameters were converted to multimode scattering matrices using (6)-(9). The attenuation coefficient α could thereafter be calculated, using the eigenvalue problem in (10) together with (3). The latter procedure will later be referred to as the HOM method.

The simulation of the original structure presented in [3] used perfect electrical conductor (PEC) and a non-complex relative permittivity of $\epsilon_r = 4.5$. As simulation difficulties occurred, the simulations for the novel structures presented in this article were implemented using non-ideal materials, which were annealed copper and lossy FR4 substrate with $\epsilon_r \approx 4.5$. These materials are standardized in CST. For every figure presented in this article, cyan material signifies substrate and gray material signifies metal.

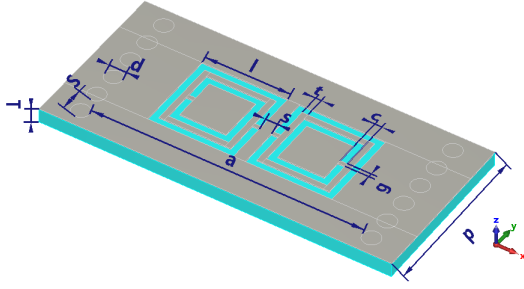


Fig. 2. Top of CSRR unit-cell from [3], used both in 1-sided non-symmetric and 2-sided glide-symmetric variant.

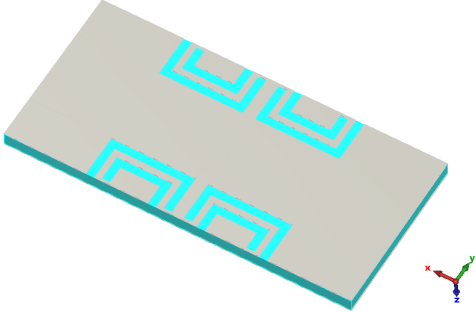


Fig. 3. Bottom of 2-sided glide-symmetric CSRR unit-cell from [3], with same dimensional parameters as Fig. 2. The bottom resonators are not prevalent in the 1-sided structure.

A. Validation of simulation technique using known unit-cell

As the structure of interest in this article is based on [3, Fig. 1], the presented simulation technique was validated through a recreation of the 1-sided and the 2-sided glide symmetric unit-cells. The phase constant β and the cut-off frequencies of the unit-cell's modes from [3, Fig. 7] could thereafter be compared with our recreation in CST using eigenmode solver. When referring to Fig. 2 and 3, the dimensions that were used are presented in Table I.

The simulation software used in [3] was Ansys HFSS, and the simulation software used in this article was CST. The eigenmode solver in Ansys HFSS can be used with open boundary conditions, which CST-ES is unable to do.

TABLE I
STARTING DIMENSIONS IN THE QUALITATIVE PARAMETRIC STUDY OF A KNOWN STRUCTURE FROM [3].

Dimensional variable	Value [mm]
a	12.4
c	0.32
s	0.54
t	0.26
g	0.18
T	0.63
p	7.55
d	0.8
S	1.2
l	3.92

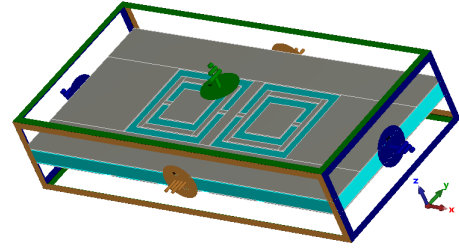


Fig. 4. Boundary conditions used for CST-ES, where green represents $E_{\tan} = 0$, blue represents $H_{\tan} = 0$ and yellow represent periodic bounds.

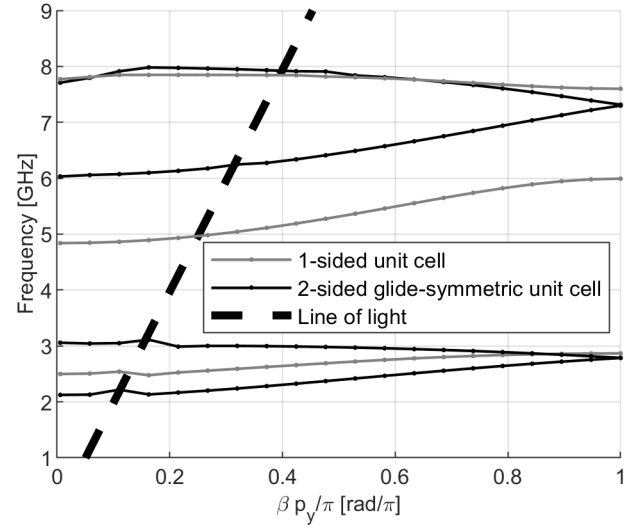


Fig. 5. The normalized β with respect to frequency from CST-ES using two recreated structures from [3]. Dimensions used are presented in Table I

Boundary conditions in CST were set such that the $E_{\tan} = 0$ for z -bounds, $H_{\tan} = 0$ for x -bounds and periodical in the y -direction, see Fig. 4. The boundary conditions in the x - and z -planes by themselves are equivalent to a parallel plate waveguide [8, p. 102]. As the boundary condition used for CST-ES are not open, extra modes were prevalent in the results similar to modes for a parallel plate waveguide, which had to be manually removed as they are not physical for the actual unit-cell. When these extra modes were manually removed, Fig. 5 was obtained. Despite some dislocated values around the line of light, Fig. 5 is very similar to [3, Fig. 7], which validates our method in CST-ES.

The attenuation coefficient α was obtained for the 2-sided glide-symmetric variant using the HOM method, as shown in Fig. 6. The number of excited modes were set to $M = 10$, which yielded a transfer matrix of size 20×20 as stated in section II-F. A comparison of β between Fig. 5 and 6 validates the HOM method, as the results show similar values for β .

B. Adjusted 2-sided glide-symmetric structure for FR4 substrate

In order to obtain a filter with the given stopband, a parametric study had to be conducted. This was done by manually changing parameters in the unit-cell presented in Fig. 2 and 3. The study was conducted manually due to the fact that

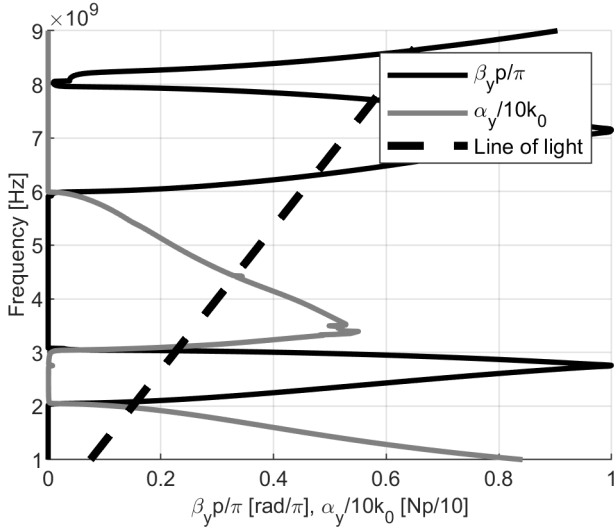


Fig. 6. The phase- and attenuation coefficients for the 2-sided glide-symmetric structure from [3] using the eigenvalue problem presented in this thesis.

the group desired a qualitative understanding of the correlation between certain parameters and the following effect on the filter's properties. The defined starting point was provided in [3] and were initially set as in Table I.

The substrate that was used was of the FR4 type with a thickness of $T = 1.5$ [mm], and a relative permittivity of $\epsilon_r = 4.5$. The qualitative parametric study was done such that the stopband and cut-off frequencies for the original structure was recreated as correct as possible. The dimensions for the new 2-sided glide-symmetric structure for FR4 substrate are presented in Table II.

Following the same procedure as before, the phase coefficient β was calculated using CST-ES. The first four modes are plotted in Fig. 7, showing cut-off frequencies for the stopband around 3 [GHz] and 6 [GHz] as in the original glide-symmetric structure, see section II-D. Similar to the previously simulated structure using CST-ES, non-physical modes were prevalent in the results due to non-open boundary conditions which were manually removed.

The propagation coefficient γ was obtained using the HOM-method for this novel structure. The number of modes that

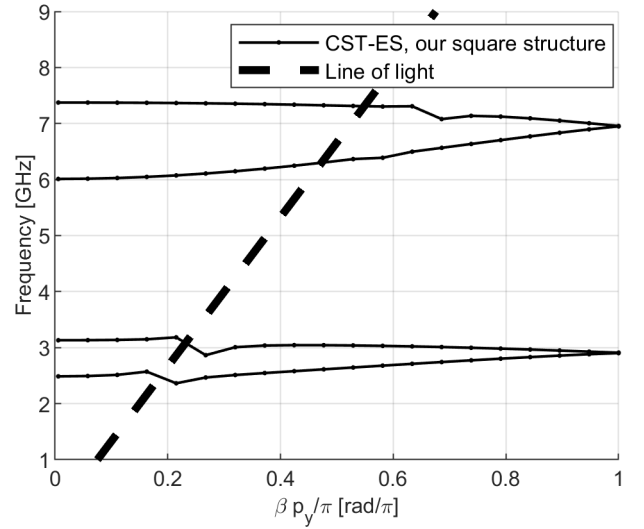


Fig. 7. The first four modes for the adjusted implementation of the 2-sided glide-symmetric structure with dimensions from Table II.

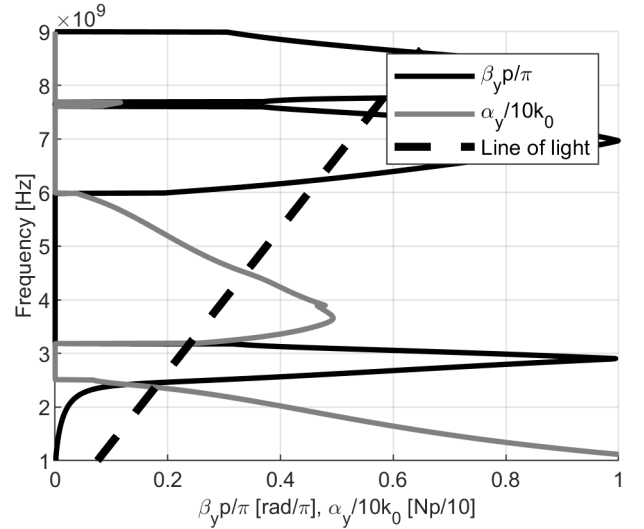


Fig. 8. The phase- and attenuation coefficient for the novel square glide-symmetric CSRR.

TABLE II
FINAL DIMENSIONS AFTER QUALITATIVE PARAMETRIC STUDY, USED ONLY FOR 2-SIDED GLIDE-SYMMETRIC STRUCTURE WITH FR4 SUBSTRATE.

Dimensional variable	Value [mm]
a	12.67
c	0.48
s	0.66
t	0.39
g	0.27
T	1.5
p	11.25
d	1.19
S	1.79
l	5.84

were excited during the parametric study was $M = 14$, which yielded a 28×28 transfer matrix. This simulation gave results for the structure's attenuation, as shown in Fig. 8. The figure shows some non-physical results, as both α and β are non-zero for frequencies below 2.5 [GHz]. As this does not occur for the stopband, when a comparison between this results and the result from CST-ES, one can analyze that the same cut-off frequencies are obtained for the modes. One can also see that the maximum attenuation is identical to the 2-sided glide-symmetric structure in Fig. 6 presented in [3].

C. 2-sided glide-symmetric circular CSRR SIW

Using the same FR4 substrate as implemented in the former section, a novel 2-sided glide-symmetric structure with CSRR in SIW was simulated using circular CSRR instead of square ones. This was done using the same dimensions as presented

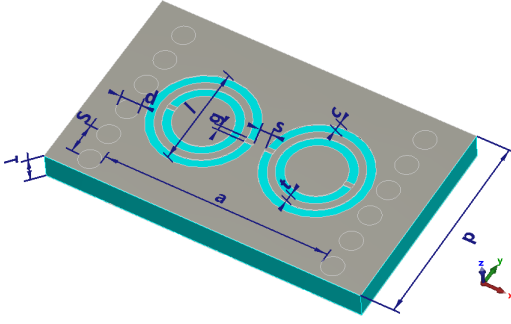


Fig. 9. The top of the 2-sided glide-symmetric circular CSRR SIW unit-cell structure with dimensional definitions. Values for each dimension is given in Table II.

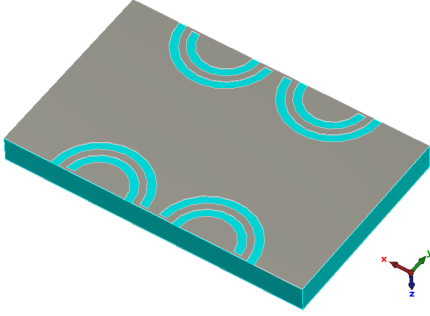


Fig. 10. The bottom of the 2-sided glide-symmetric circular CSRR SIW unit-cell structure. The structure dimensions are defined as in Fig. 9.

in Table II, which lead to a comparison how the circular resonators compares to the square resonators regarding cut-off frequencies and attenuation. The structure is presented in Fig. 9 and 10.

The phase coefficient β from CST-ES was obtained, and the results are shown in Fig. 11. The cut-off frequencies has been shifted compared to the square CSRR structure with same geometrical constraints. The circular CSRR structure has higher cut-off frequencies, and a stopband between 4 [GHz] and 6.5 [GHz]. The square CSRR structure has stopbands between 3 [GHz] and 6 [GHz]. A narrower stopband is obtained with the circular CSRR structure.

Using the HOM-method for the circular CSRR structure, Fig. 12 was obtained. The amount of modes that were excited was $M = 10$, which yielded a 20×20 transfer matrix as stated in section II-F. A comparison between this result and Fig. 8 yields that the ring CSRR structure has a lower maximum attenuation than the quadratic CSRR structure. The respective maximum attenuation coefficient in each stopband were obtained as $\alpha_{\text{quadratic}} = 378.6 \text{ [m}^{-1}\text{]}$ and $\alpha_{\text{ring}} = 378.0 \text{ [m}^{-1}\text{]}$. Furthermore, when the structure has the same geometrical constraints, the cut-off frequencies for the stopband are generally higher for the circular CSRR structure. The bandwidth of the stopband is also more narrow for the circular CSRR structure.

IV. SIMULATION OF COMPLETE FILTERS

Two complete filters were implemented based on the novel unit-cells proposed in this article. The complete structures

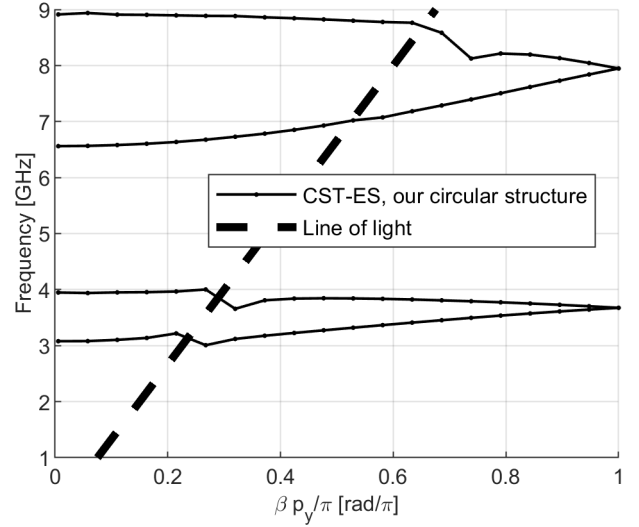


Fig. 11. The phase coefficient for the novel circular CSRR SIW for the four first modes using CST-ES.

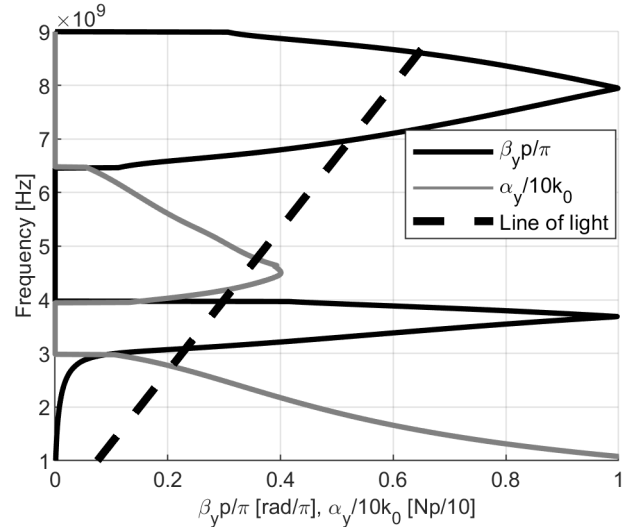


Fig. 12. The phase- and attenuation coefficient obtained using the HOM-method for the circular CSRR structure presented in Fig. 9 and 10.

with transitions to microstrips were simulated. Both filters were port-symmetrical and used eight CSRR's on top and ten CSRR's at the bottom, see Figs. 13 and 14.

A. Matching transition

The complete filters for the two proposed unit-cells are designed to be connected via two SMA coaxial connectors which requires two microstrip lines to be matched to the filter, as SMA connectors are easy to solder onto a microstrip. A transition between the repeating unit-cells and the microstrips were made using taper transitions [15]. The dimensions w and Q were parametrically studied such that the S_{11} parameters were simultaneously as low as possible for both passbands frequency ranges, see Fig. 13 and 14. This condition is important for the complete filters, as reflected waves are unwanted. The width m of the microstrip line was designed

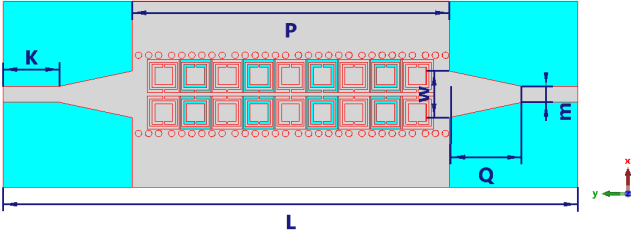


Fig. 13. The implementation of the proposed square CSRR filter of this article, showing both top and bottom resonators. Resonators with cyan contours are on top, and others are etched on the filters bottom side.

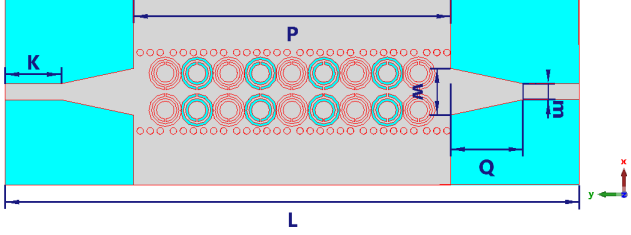


Fig. 14. The implementation of the proposed circular CSRR filter of this article, showing both top and bottom resonators. Resonators with cyan contours are on top, and others are etched on the filters bottom side.

to have a characteristic impedance of $50 \, [\Omega]$. All dimensions for the complete structures are presented in Tables III and IV. The simulated S_{11} and S_{12} are presented in Fig. 15 and 16. These results could be verified with physical implementations.

For some reason unknown to the authors, the filter designs that are proposed were unable to be matched with a taper transition such that S_{11} for the first passband was lower than $-5 \, \text{dB}$. This mismatching may be avoided, as the transition from the unit-cells to microstrip line may be unnecessary in a physical implementation. When choosing the dimensions

TABLE III

DIMENSIONS FOR THE COMPLETE FILTER STRUCTURE IMPLEMENTING SQUARE CSRR'S, TAPER TRANSITIONS AND MICROSTRIP LINES

Dimensional variable	Value [mm]
K	10
P	56.25
L	101.92
w	8.24
Q	12.84
m	2.84

TABLE IV

DIMENSIONS FOR THE COMPLETE FILTER STRUCTURE IMPLEMENTING CIRCULAR CSRR'S, TAPER TRANSITIONS AND MICROSTRIP LINES

Dimensional variable	Value [mm]
K	10
P	56.25
L	95.41
w	6.35
Q	9.58
m	2.844

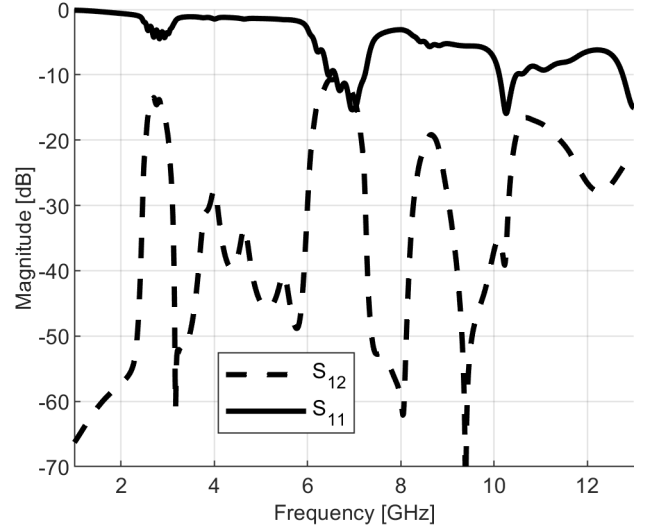


Fig. 15. Simulated S-parameters for the complete structure consisting of square CSRRs, see Fig. 13.

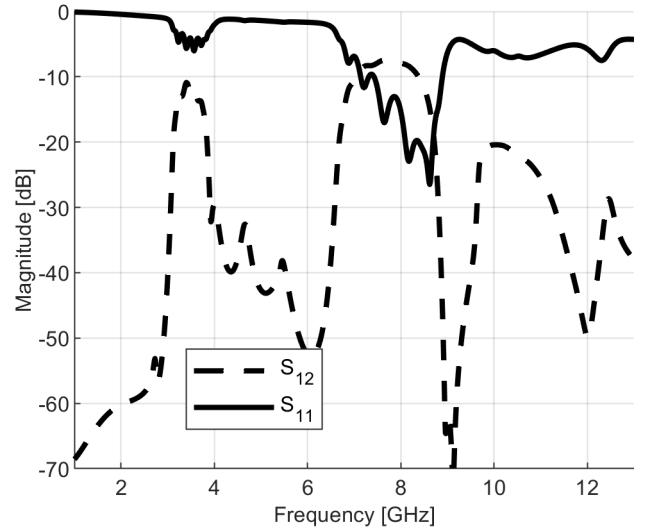


Fig. 16. Simulated S-parameters for the complete structure consisting of circular CSRRs, see Fig. 14.

w and Q from the parametric study, local minimums were chosen where S_{11} was low for both stopbands, although not necessarily below a certain limit in dB. The problem with mismatching in the first passband was also prevalent in the original design when using taper transitions [3].

Another defect visible in Fig. 15 and 16 is that each cut-off frequency has been increased slightly compared to the unit-cells dispersion diagrams. The reason for this is also unknown to the authors, although an obvious consequence of the addition of the transitions.

V. CONCLUSIONS

Two novel unit-cells for bandgap filters were proposed. The HOM method was used to simulate the unit-cells attenuation, which commercial simulation software is unable to do. Complete structures containing the unit-cells were simulated,

although with inaccurate results as a result of a bad matching transition.

When given the same geometrical constraints, using circular CSRRs is proven to have higher cut-off frequencies for the filters stopband compared to implementing square CSRRs. Implementing circular CSRRs is although also proven to shorten the bandgap. Both implementations are shown to yield the same maximum attenuation in their stopbands.

The authors noted that the lower permittivity $\varepsilon_r = 4.5$ for the FR4 substrate in general dictated the unit-cells geometries to be larger compared to the original structure [3]. Therefore, a conclusion might be drawn that higher permittivity may be beneficial for bandgap filter designs, as higher relative permittivity enables the structure to be smaller.

Future work within this context may be to construct complete filters based on the proposed unit-cells with correct matching, or without a transition to microstrip lines at all. Despite of the bad matching of the complete structures shown in this article, the filters may be combined with a microwave amplifier. The difference of S_{12} between the passbands and the stopbands is however still significant, around 20 dB, and signals could therefore be amplified to create a practical stopband filter.

ACKNOWLEDGMENT

The authors would like to extend sincere thanks to Freysteinn Vidar Vidarsson, Chen Mingzheng and Oscar Quevedo Teruel for providing guidance, assistance and their knowledge to the project.

REFERENCES

- [1] 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, Dec. 2020.
- [2] I. Shahid, D. Thalakituna, D. K. Karmokar, S. J. Mahon, and M. Heimlich, "Periodic structures for reconfigurable filter design: A comprehensive review," *IEEE Microwave Magazine*, vol. 22, no. 11, pp. 38–51, Nov. 2021.
- [3] J. Martínez, A. Coves, F. Mesa, and O. Quevedo-Teruel, "Passband broadening of sub-wavelength resonator-based glide-symmetric siw filters," *AEU - International Journal of Electronics and Communications*, vol. 125, p. 153362, Oct. 2020.
- [4] A. Orlandi, B. Archambeault, F. de Paulis, and S. Connor, *Removable EBG Common Mode Filters*. Hoboken, NJ, USA: Wiley, 2017, pp. 165–198.
- [5] J. Baena, J. Bonache, F. Martin, R. Sillero, F. Falcone, T. Lopetegi, M. Laso, J. Garcia-Garcia, I. Gil, M. Portillo, and M. Sorolla, "Equivalent-circuit models for split-ring resonators and complementary split-ring resonators coupled to planar transmission lines," *IEEE Transactions on Microwave Theory and Techniques*, vol. 53, no. 4, pp. 1451–1461, Apr. 2005.
- [6] D. Gupta, K. Chaurasia, A. Upadhyay, and P. K. Singhal, "Csrr based low pass microstrip filter using stepped impedance," in *2014 International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom)*, Nov. 2014, pp. 35–39.
- [7] D. Deslandes and K. Wu, "Integrated microstrip and rectangular waveguide in planar form," *IEEE Microwave and Wireless Components Letters*, vol. 11, no. 2, pp. 68–70, Feb. 2001.
- [8] D. M. Pozar, *Microwave engineering*, 4th ed. Hoboken, NJ, USA: Wiley, 2012.
- [9] D. K. Cheng, *Field and wave electromagnetics*, 2nd ed. Harlow, Essex, England: Pearson, 2014, pp. 520–599.
- [10] R. E. Collin, *Field Theory of Guided Waves*. Hoboken, NJ, USA: Wiley, 1991, vol. 2, pp. 605–643.
- [11] F. Mesa, G. Valerio, R. Rodríguez-Berral, and O. Quevedo-Teruel, "Simulation-assisted efficient computation of the dispersion diagram of periodic structures: A comprehensive overview with applications to filters, leaky-wave antennas and metasurfaces," *IEEE Antennas and Propagation Magazine*, vol. 63, no. 5, pp. 33–45, Oct. 2021.
- [12] G. Valerio, S. Paulotto, P. Baccarelli, P. Burghignoli, and A. Galli, "Accurate bloch analysis of 1-d periodic lines through the simulation of truncated structures," *IEEE Transactions on Antennas and Propagation*, vol. 59, no. 6, pp. 2188–2195, Jun. 2011.
- [13] A. Hessel, M. H. Chen, R. Li, and A. Oliner, "Propagation in periodically loaded waveguides with higher symmetries," *Proceedings of the IEEE*, vol. 61, no. 2, pp. 183–195, Feb. 1973.
- [14] B. A. Mouris, A. Fernández-Prieto, J. L. M. d. Río, R. Thobaben, J. Martel, F. Mesa, F. Medina, and O. Quevedo-Teruel, "Glide symmetry applied to printed common-mode rejection filters," *IEEE Transactions on Microwave Theory and Techniques*, vol. 70, no. 2, pp. 1198–1210, Feb. 2022.
- [15] D. Deslandes and K. Wu, "Integrated microstrip and rectangular waveguide in planar form," *IEEE Microwave and Wireless Components Letters*, vol. 11, no. 2, pp. 68–70, Mar. 2001.

Design of a Leaky-wave Antenna Based on Goubau Line for Imaging Applications

Roger Bock Filella

Abstract—The project main purpose is to design a prototype of a leaky-wave antenna (LWA) which will serve as a basis to future manufacturing of a more advanced product with potential use for an imaging radar system. The antenna consists of Goubau line corrugations with rectangular radiation patches. Both configurations with longitudinal symmetric and glide symmetric corrugations have been analyzed, paying special attention to their dispersion diagrams. A parametric study of the dimensions of the structure has been conducted as well. Radiation patches have been added with different offsetings between each other, in order to study the effect of the separation between patches to the radiation pattern. The frequency beam scanning characteristic of the antenna is also shown. This antenna operates at a center frequency of 9.4 GHz, with an efficiency of 80 %. It scans over an approximate angle width from -50 to 50° in a frequency range from 8.5 to 10.2 GHz. It has been modeled using *CST Microwave Studio* and *Matlab*.

Sammanfattning—Det huvudsakliga målet för projektet är att designa en prototyp av en läck-vågsantenn (LWA) som ska verka som en grund för framtida tillverkning av mer avancerade produkter, med potentiell användning för radarsystem för bildtagning. Antennen består av en Goubau linje veckning med rektangulära strålningsflikar. Båda konfigurationerna med longitudinell symmetri och glid symmetri har analyserats. Förutom detta har en parameterstudie av dimensionerna av strukturen utförts. Strålningsflikar har blivit tillagda med olika förskjutningar mellan varandra, för att studera effekten av separationen mellan flikarna och strålningsmönstret. Egenskaperna hos antennens frekvensberoende skanning visas också. Denna antenn fungerar vid en centerfrekvens på 9.4 GHz, med en effektivitet på 80%. Den skannar över en ungefärlig vinkel från -50 till 50° över ett frekvensspann från 8.5 till 10.2 GHz. Detta har modellerats med *CST Microwave Studio* och *Matlab*.

Index Terms—leaky-wave antenna, radar systems, Goubau Line, glide symmetry, dispersion diagram, endfire, broadside.

Supervisors: Shiyi Yang, Qiao Chen, Oscar Quevedo-Teruel, Juan Manuel Rius Casals.

TRITA number: TRITA-EECS-EX-2022:150

I. INTRODUCTION

Radar and imaging systems require agile scanning capabilities. The scanning can be achieved in multiple ways, such as mechanical rotation or modifying the progressive phase of the antenna elements in an array. A commonly used technique is frequency scanning, as it provides a flexible control over the range resolution and scanning performance. The most straightforward solution to get frequency scanning is to use LWAs, as they present simple implementation and feeding, while providing high efficiency. In terms of cost, Goubau

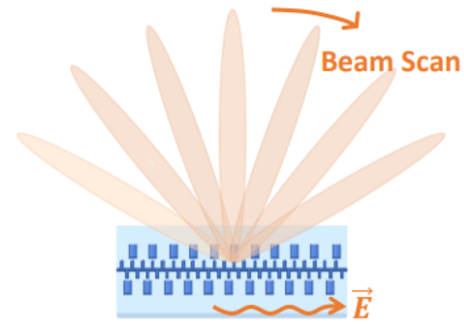


Fig. 1. Frequency beam scanning characteristic of a leaky-wave antenna.

line is an effective choice featuring easy integration and low-loss transmission [1]. In fact, LWAs are a class of antennas that use a travelling wave in a guiding structure as the main radiation mechanism. [2]. Fig.1 reflects the inherent frequency beam scanning property of leaky-wave antennas. Generally, the wave travelling in the guiding structure is a fast wave, with a phase constant β smaller than the free-space wavenumber k_0 . Therefore, the travelling wave is radiating and leaks power continuously while it propagates. Depending on the guiding structure, leaky wave antennas can be classified as uniform (the geometry does not change along the length of the structure) and periodic. Our design will consist of a one dimensional periodic leaky-wave antenna, in which a repetition of periodic discontinuities along the propagation direction are added in order to make a slow wave radiate. The radiation characteristics preferred for a LWA mostly depend on the application. For surveillance a high directivity and a wide angle scanning prove to be essential, as it occurs in [3], where a frequency scanning antenna is used for an automotive radar. Imaging systems usually prefer higher frequency bandwidth, as it translates to a better spatial resolution. This is the case in [4], where multidirectional leaky-wave scanning allows to synthesize images of 3D objects placed in front of the antenna, what can be further used for remote sensing. Chao in [5] takes profit of the beam scanning properties of LWAs for contactless detection of vital signs. As shown in Fig.2 The designed system uses scanning antennas in the transmitting and receiving paths to implement a Doppler radar module capable of measuring respiration and heartbeat rates.

Mainly, periodic LWA are intended to enhance the angle width range and the scanning rate, defined as the inverse of the necessary frequency bandwidth to scan the entire angle width, ($^\circ$ / Hz). Previous works have also addressed this

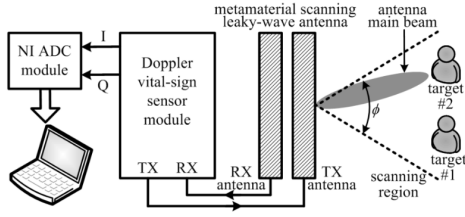


Fig. 2. Application scenario of the noncontact vital-sign radar sensor with metamaterial-based scanning leaky-wave antennas. Source: [5].

specific topic. For instance, in [6] a backward radiating LWA based on glide symmetric holey waveguide technology is reported, which includes a triangular metasurface prism so as to enhance the scanning rate. However, the unit cell of this periodic LWA is based on a closed structure, unlike the case of study in this report. In [1] a periodic LWA featuring Goubau line is studied, emphasizing on the improvement of the radiation characteristics when transforming the periodic radiation patches in an asymmetrical fashion. The guiding structure of [7] is the one implemented for the antenna design in this report. Concretely, it studies both the longitudinal symmetric and glide symmetric configuration of stubs in a transmission line and their correspondent transmission and dispersion characteristics, essential for acquiring a high scanning rate. The work reported in [8] uses this same guiding structure to design a LWA, focusing on the trade-off between scanning rate enhancement and radiation efficiency and presenting an antenna prototype capable of scanning through the broadside. This work will serve as a basis for the antenna design reported in this paper.

II. THEORY

A. Goubau Line

Goubau line is a groundless single conductor transmission line which presents low loss and easy integration [1]. Compared to microstrip line or coplanar waveguide, which are used as wave guiding structures for many microwave devices, Goubau line features an enhanced radiation efficiency. This is due to the fact that, unlike microstrip and coplanar waveguide, it does not suffer from such severe losses generated in the dielectric and ground planes. This issue becomes even more problematic at high frequencies, at which the dielectric losses increase significantly.

In Fig. 3 an illustration of a planar Goubau line is depicted. It consists of a substrate with dielectric constant ϵ_r and thickness h and a conductor metallic strip of width w and thickness t . The fundamental mode propagating along the Goubau line is a slow wave, that does not radiate by itself. Radiation can be obtained by either introducing periodic discontinuities along the structure or exciting higher order modes that support fast wave propagation.

B. Radiation of Leaky-Waves

Because of the leakage of power, the propagation wavenumber on the guiding structure of a LWA is complex and can

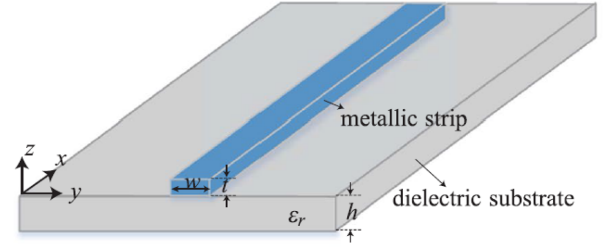


Fig. 3. Planar Goubau line. Source: [1]

be expressed in the form of $k_z = \beta - j\alpha$, with z being the propagation direction, β the phase constant and α the attenuation due to leakage as the wave propagates. As a general rule of thumb, the antenna beamwidth is proportional to the attenuation constant whereas the beam direction is related to the phase constant [9]. Assuming x to be the transversal direction with respect to the structure length [10], the electric field can be expressed as

$$E_y(x, z) = Ae^{-jk_z z} e^{-jk_x x} \quad (1)$$

where A is the complex amplitude and k_z and k_x are the wavenumbers in the longitudinal and transversal directions. They must satisfy

$$k_0^2 = k_z^2 + k_x^2 = (\beta - j\alpha)^2 + (\beta_x - j\alpha_x)^2 \quad (2)$$

where $k_0 = 2\pi f/c$ is the wavenumber in free space. Equating imaginary parts leads to

$$\alpha\beta + \alpha_x\beta_x = 0 \quad (3)$$

Hence, assuming a forward wave propagation in the longitudinal direction ($\alpha > 0$ and $\beta > 0$) in order to have outward propagation, $\beta_x > 0$, what leads to $\alpha_x < 0$, meaning that the outward wave is exponentially increasing. Therefore, a forward wave decaying in the longitudinal direction due to leakage loss must increase in the transversal direction what explains the principle of radiation of leaky waves. The same reasoning can be applied for a backward propagating wave. The direction of the beam [2] can be obtained as the angle formed by the phase vector $\beta = \beta_x \hat{x} + \beta_z \hat{z}$. Defining θ as the angle with respect to the perpendicular direction, as shown in Fig. 4, it can be approximated when the attenuation is small by

$$\theta(f) = \arcsin\left(\frac{\beta}{k_0}\right) \quad (4)$$

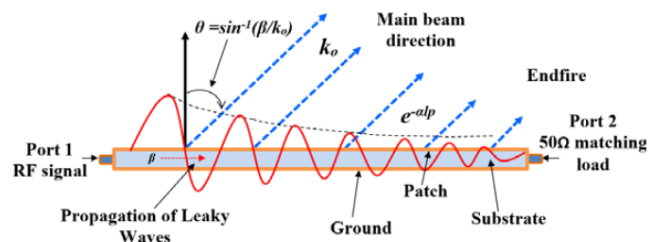


Fig. 4. Propagation of a uniform leaky wave. Source: [11].

C. Floquet's Theorem

A 1D periodic structure consists of a repetition of periodic discontinuities along the propagation direction, each of them separated a distance equal to the period p . The mutual interaction between periodic elements is responsible for mode coupling resonances, which might result in the formation of stopbands and pass bands. Because of the periodicity, the field of the periodic structure can be expressed in the form of a Floquet mode expansion [12]. The Floquet theorem states that any field propagating in the z direction will satisfy

$$\mathbf{E}(x, z + p) = e^{-jk_{z0}p} \mathbf{E}(x, z) \quad (5)$$

where $k_{z0} = \beta_0 - j\alpha$ is the wavenumber of the fundamental space harmonic. In fact, the electric field consists of a series expansion of an infinite number of travelling waves, labelled as $n = 0, \pm 1, \pm 2, \dots$; the so called space harmonics. The expression of the electric field results in

$$\mathbf{E}(x, z) = \sum_{n=-\infty}^{\infty} \mathbf{A}_n(x) e^{-j(k_{z0} + \frac{2\pi}{p}n)z} \quad (6)$$

with $\mathbf{A}_n(x)$ being the complex amplitude of each space harmonic, for $n = 0, \pm 1, \pm 2, \dots$. For each of them, a wavenumber can be defined as $k_{zn} = k_{z0} + \frac{2\pi}{p}n = \beta_n - j\alpha$. Therefore, all of them have equal attenuation α but different phase constants

$$\beta_n = \beta_0 + \frac{2\pi}{p}n \quad (7)$$

This means that any harmonic in the fast-wave region radiates independently from the others in the direction specified by Eq. 4, which for a given value of n particularizes as:

$$\theta_n(f) = \arcsin\left(\frac{\beta_n}{k_0}\right) \quad (8)$$

D. Dispersion Diagram

The dispersion diagram is a representation of the phase propagation constant $\beta(f)$. In the case of Floquet wave harmonics it can be reduced to the Brillouin zone defined as $-\pi/p \leq \beta_n \leq \pi/p$ due to its periodicity every $\beta p = 2\pi$. Fig. 5 shows the information provided by the dispersion diagram for the case of a periodic LWA, represented in terms of normalized frequency $k_0 p / \pi$ in the ordinate and normalized phase constant $\beta p / \pi$ in the abscissa. It allows to determine the space harmonics inside the fast-wave region ($|\beta_n| < k_0$) and their correspondent beam angles θ_n at different frequencies, according to Eq. 8. The dashed line $|\beta| = k_0$ is known as the line of light and represents the boundary between slow wave and fast wave propagation.

Other information that can be extracted from the dispersion diagram is the scanning rate [8]. As it is defined as the scanned angle divided by the frequency bandwidth, the lower the slope of the dispersion curve, the higher the scanning rate. The number of beams corresponds to the number of space harmonics located inside the fast wave region. In order to have only one radiation beam [2], it is convenient to let the harmonic $\beta_{-1} = \beta_0 - 2\pi/p$ fall inside the fast wave region, while the other harmonics remain bound. Therefore, the $n = -2$ space harmonic must remain a slow backward wave ($\beta_{-2} < -k_0$), whereas

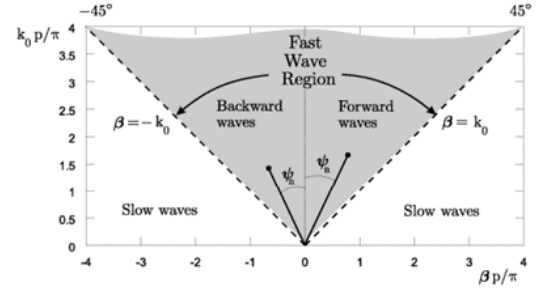


Fig. 5. Dispersion diagram of a 1D periodic structure. Source: [12].

the fundamental space harmonic is a slow forward wave ($\beta_0 > k_0$) as the $n = -1$ space harmonic scans in frequency. Harmonic coupling is responsible for the appearance of stopbands [12]. A stopband is a frequency region in which the propagation phase constant remains unaltered. The interactions between backward and forward travelling space harmonics might give rise to two different phenomena: open and closed stopbands. The open stopband occurs when two spatial harmonics couple in a region where one other space harmonic is radiative. For instance, at the broadside direction ($\theta_{-1} = 0^\circ$), $\beta_{-1} = 0$ corresponding to $\beta_0 p = 2\pi$, the forward harmonic $n = 0$ couples with the $n = -2$ backward harmonic ($\beta_0 = -\beta_{-2}$) within the region where the β_{-1} harmonic radiates. A perfect standing wave is created inside the structure and no radiation takes place. From another point of view, the separation between periodic elements becomes $p = \lambda$ and the reflections add back to the source setting up a standing wave and producing an abrupt transition of the dispersion curve. In order to obtain radiation to the broadside, different techniques can be applied. In this study two lines of radiation patches offset from one another will be used to overcome this issue. The closed stopband is also generated due to coupling between different space harmonics. However, this coupling takes place outside the fast wave region, where none of the space harmonics radiate. For instance, at $\beta_0 p = \pi$ the forward wave $n = 0$ couples with the backward wave $n = -1$ inside a bound region. In this case resulting wave does not produce leakage and the fields in the transversal direction decay exponentially, experiencing a degradation of the radiation efficiency. Fig. 6 provides a graphical explanation of the aforementioned stopband phenomena.

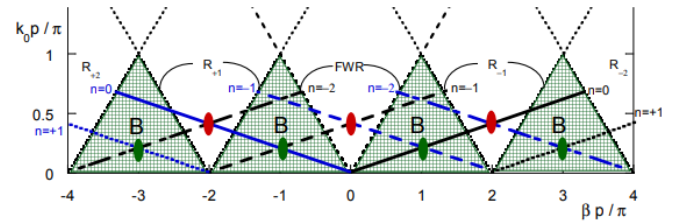


Fig. 6. Extended dispersion diagram of a 1D periodic structure. R_n represent the radiation regions for the different $n = 0, \pm 1, \pm 2, \dots$ harmonics, while B represent the bound regions. The open stopband points are represented in red while the closed stopband points are represented in green. Source: [12].

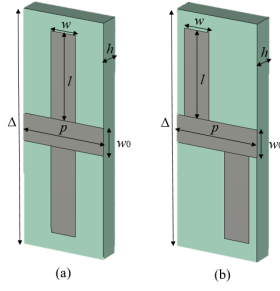


Fig. 7. Unit cell schemes: (a) Longitudinal symmetric and (b) Glide symmetric Goubau line corrugations.

TABLE I
DEFAULT DIMENSIONS OF THE UNIT CELLS

Parameter	Description	Default value (mm)
p	Modulation period	2.5
l	Length of the corrugation	2.5
w	Width of the corrugation	0.76
w_0	Width of the transmission line	0.8
h	Thickness of the substrate	0.635
Δ	Width of the substrate	40

III. BUILDING BLOCKS

A. Unit Cell

In this section two different unit cells will be studied. Fig.7(a) and Fig.7(b) show the both schemes under study, featuring longitudinal and glide symmetric Goubau line corrugations respectively. The substrate used is ROGERS RO3010, with $\epsilon_r = 10.2$ and $\tan \delta = 0.0022$. The dimensions specified in Table I will be considered the default dimensions for both schemes.

The dispersion diagrams of the different unit cells have been obtained using the *CST Microwave Studio Eigenmode Solver*. Periodic boundaries are imposed along the longitudinal direction of the transmission line. As this solver does not support open boundaries, PEC boundaries are imposed in the other two dimensions, leaving a spacing of B between the boundary and the unit cell. In order to ensure convergence of the results, a parametric study of B is required, as shown in Fig. 8. As a result, to guarantee convergence, the minimum required spacing is set to $B = 68\text{mm}$.

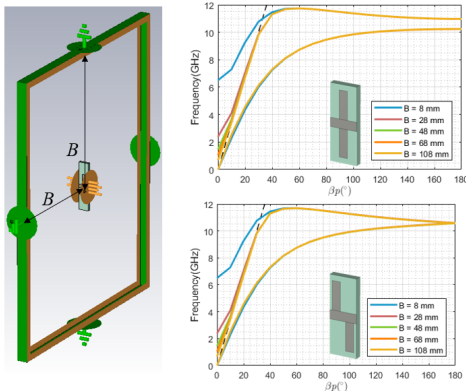


Fig. 8. Study of the convergence of the dispersion diagram.

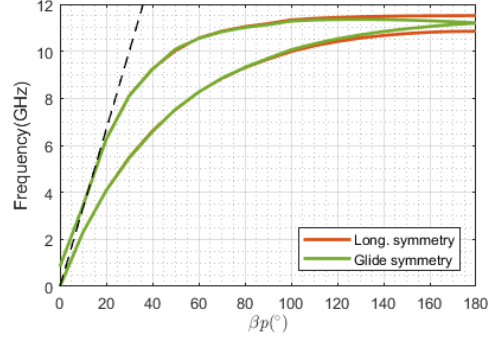


Fig. 9. Dispersion diagram of the two unit cells using the default dimensions.

The dispersion diagram of the first two modes for both structures is shown in Fig. 9. The dashed line represents the line of light. Both dispersion diagrams fall outside the fast wave region, as no radiation patches have been included so far. As it can be appreciated, the glide symmetric configuration closes the stopband near the $\beta p = \pi$ boundary, which corresponds to a closed stopband of the periodic structure. The region close to this boundary will provide a very high scanning rate, as the slope of the dispersion curve is nearly flat. Therefore, if one is able to move this region to the broadside with the inclusion of radiation patches a very high scan rate can be achieved. In contrast, the longitudinal symmetric design does not close the stopband. This is the reason why reflections will become more severe, degrading the radiation efficiency significantly. However, if the dispersion curve is not shifted from the $\beta p = \pi$ point to the broadside both schemes will present a similar performance.

A parametric study of the unit cell dimensions is carried out in Fig. 10. In Fig. 10(a) the modulation period is varied. As it decreases the scanning rate is enhanced, because the dispersion curve presents a smaller slope. In the longitudinal symmetric case, the stopband between both modes is reduced as well. The same effects occur in Fig. 10(c), in which the width of the transmission line is altered. Finally, decreasing the corrugation length in Fig. 10(b) implies that the dispersion curve is shifted upwards. The plot representing the variation of the dispersion curve when w varies has not been included due to the changes in the dispersion curve being barely noticeable. Regarding the dimensions of the substrate, the dispersion curve converges as both its width and its thickness increase, as it can be seen in Fig. 11.

B. Feeding, Transition and Tapering

In order to connect the antenna to other circuit elements the design of the feeding needs to be implemented. For our design, coplanar waveguide (CPW) represents a reliable choice due to its simple realization and low dispersion [13]. Its dimensions are tabulated in Table II and have been chosen to ensure good matching at a reference impedance of 50Ω . Fig. 12 shows the S-parameters of the simulated coplanar waveguide. s_{11} measures the fraction of power inserted from one port that gets reflected whereas s_{21} measures the fraction of power that is transmitted all the way to the end of the structure. As it can

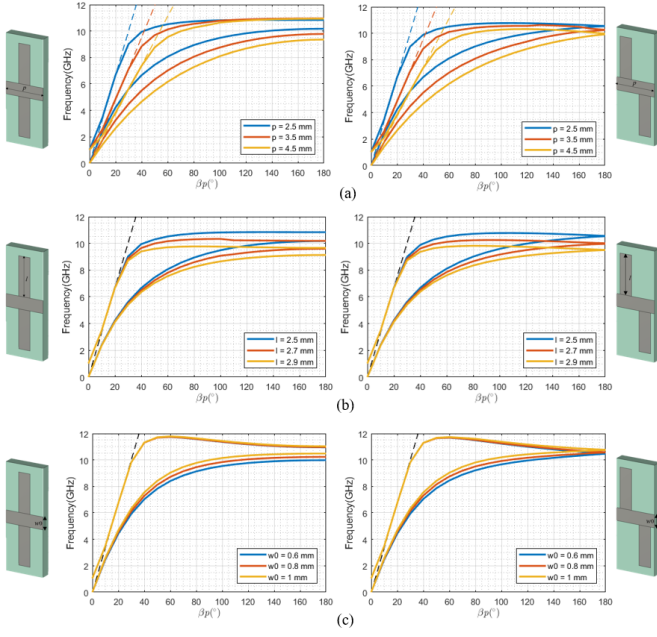


Fig. 10. Parametric study of the unit cell dimensions: (a) Variable p , (b) Variable l and (c) Variable w_0 .

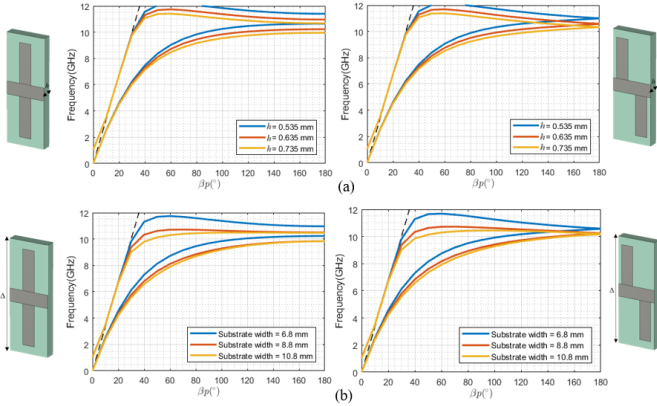


Fig. 11. Parametric study of the substrate dimensions: (a) Variable h , (b) Variable Δ .

be appreciated s_{21} remains around 0 dB, which means that the energy flowing through port 1 is transmitted all the way to port 2. The s_{11} is smaller than -40 dB so the reflections at the input can be neglected.

A CPW to microstrip transition is required to make the connection between the CPW and the Goubau line. An exponential transition of length $l_{tran} = 40$ mm has been used. The initial dimensions of the transversal cross section of the transition

TABLE II
DIMENSIONS OF THE CPW

Parameter	Description	Value (mm)
l_{CPW}	Length of the CPW	10
s	Separation between conductors	0.44
w	Width of the central strip	0.8
t	thickness of the strip	0.1

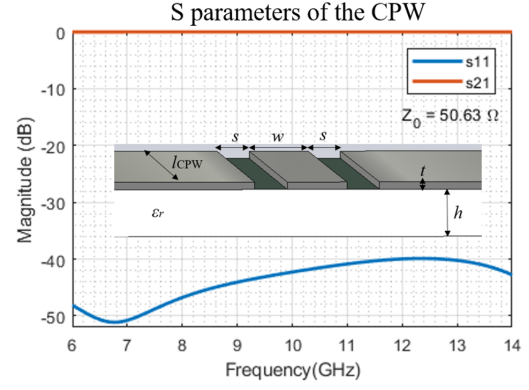


Fig. 12. S-parameters of the CPW used to feed the antenna

need to coincide with the ones of the CPW. In Fig. 13 there is a representation of the transition, as well as the resulting S-parameters. In this case, the reflections at the input are more significant but still negligible, as $s_{11} < -15$ dB.

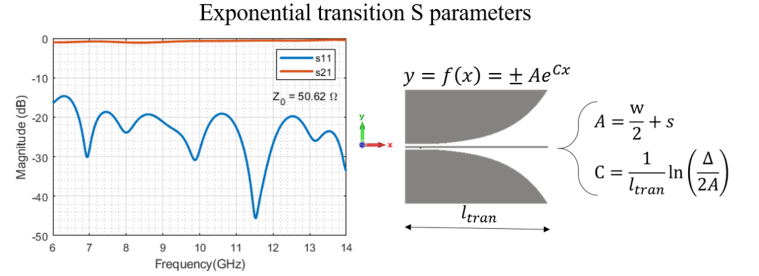


Fig. 13. S-parameters of the exponential transition and its equation.

Finally, a linear tapering is added to make the passage of the electromagnetic fields softer between the exponential transition and the Goubau line corrugations. This way the loss due to reflections is minimized. The two different tapering structures shown in Fig. 14 correspond to the longitudinal and glide symmetric schemes respectively. Fig. 15 shows the final structure of both LWA designs, including a set of rectangular radiation patches.



Fig. 14. Tapering: (a) longitudinal and (b) glide symmetric designs.

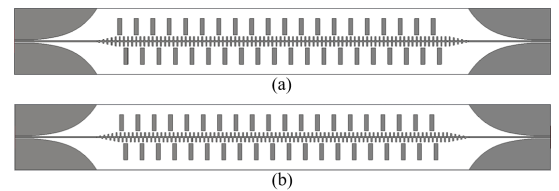


Fig. 15. Leaky-wave antenna: (a) longitudinal and (b) glide symmetric designs.

C. Radiation Patches

The dimensions of the patches are written down in Table III and shown in Fig. 17. The *offset* value will be modified in the following sections to see how it affects the different radiation parameters. The loading between patches has been set to $d = 4p$, shifting the dispersion curves in Fig. 9 around $\beta p = \pi/2$ to the broadside, thus making the harmonic $n = -1$ radiate. This can be seen in Fig. 16, where the extended dispersion diagram showing the Floquet harmonics is represented.

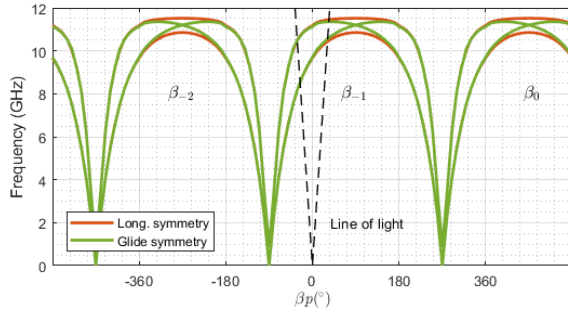


Fig. 16. Extended dispersion diagram of the two unit cells using the default dimensions.

In fact, a reduced expression for the beam direction can be obtained in this specific case:

$$\theta(f) = \arcsin \left[\frac{c}{2\pi f p} (\beta_0 p - \pi/2) \right] \quad (9)$$

from which it is deduced that the operating band is similar in both cases, going from 8.5 GHz to roughly 10.2 GHz. It sweeps an approximate angle width of 100° centered at the broadside and presents an approximate scan rate of $66.7^\circ/\text{GHz}$.

TABLE III
DIMENSIONS OF THE PATCHES

Parameter	Description	Value (mm)
l_r	Length of the patch	10
w_r	Width of the patch	2.5
g	Gap between patch and the corrugations	1
d	Patch loading distance	10
<i>offset</i>	Offset of top and bottom patches	-

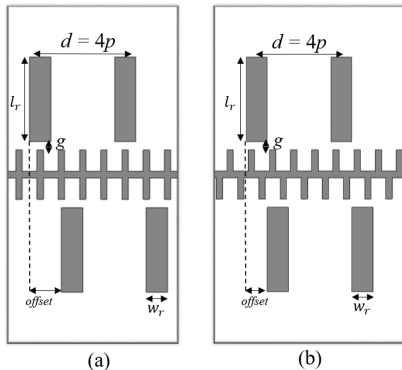


Fig. 17. Radiation patches: (a) longitudinal and (b) glide symmetric designs.

A set of patches is included at the bottom of the transmission line so as to obtain open stopband suppression, as illustrated in Fig. 18. At the open stopband the separation between radiation patches is $d = \lambda$, and hence, the reflections from two consecutive patches add in phase back to the source. The inclusion of the bottom radiators counters the effect of the reflections, as the round trip of $\lambda/2$ produces a phase shift of π , resulting in a destructive interference of the reflected signals. This corresponds to an *offset* of $\lambda/4$.

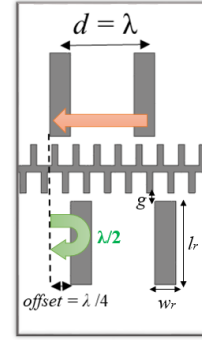


Fig. 18. Schematic justifying the open stopband suppression when a set of bottom patches is loaded

IV. RESULTS

In this section different offsets (0 mm, 1.25 mm, 2.5 mm, 3.75 mm and 5 mm) between the top and bottom radiation patches have been analyzed for both the glide and longitudinal symmetric designs. The simulations have been obtained using *CST Microwave Studio Time Domain Solver*.

A. S-parameters and Efficiency

The S-parameters are plotted in Fig. 19. Unlike the previous cases studied, the decrease of the s_{21} is justified by the antenna leakage. Large offsets provide an increased leakage of energy. Regarding s_{11} , it is maintained below -10 dB, except for the case of 0 offset within the band of interest. A peak of reflection takes place at the broadside frequency, close to 9.6 GHz. As explained before, the offset of $\lambda/4$, which corresponds to 2.5 mm, suppresses the stopband. However, an offset of 3.75 mm performs even better, possibly due to phase mismatches in the tapering and transitions of the antenna.

In terms of efficiency, the radiation efficiency improves when the offset is small whereas the total efficiency presents a similar behaviour at different offsets. This can be seen in Fig. 20. In any case the efficiencies are maintained above 80 % within the operational band (8.5 GHz to 10.2 GHz).

Comparing longitudinal and glide symmetric designs, the former performs worse at frequencies above 9.7 GHz, as it can be deduced from the s_{11} plot in Fig. 19 and the total efficiency plot in Fig. 20. Moreover, the open stopband suppression proves to be better for the glide symmetric configuration as well.

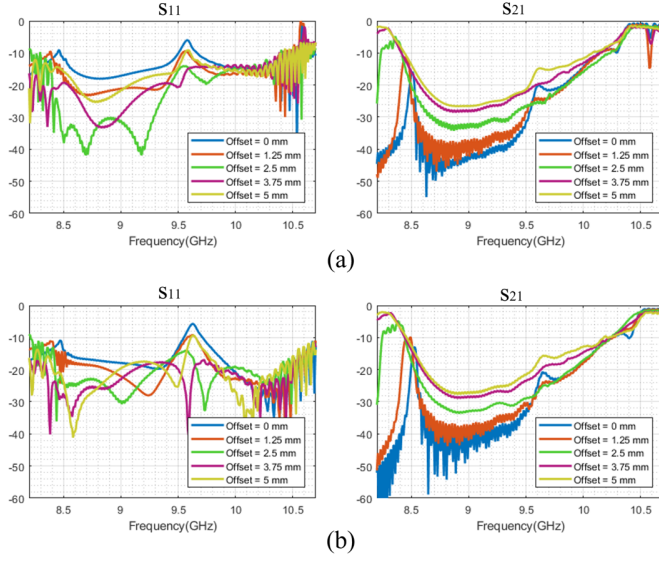


Fig. 19. S-parameters: (a) longitudinal and (b) glide symmetric designs.

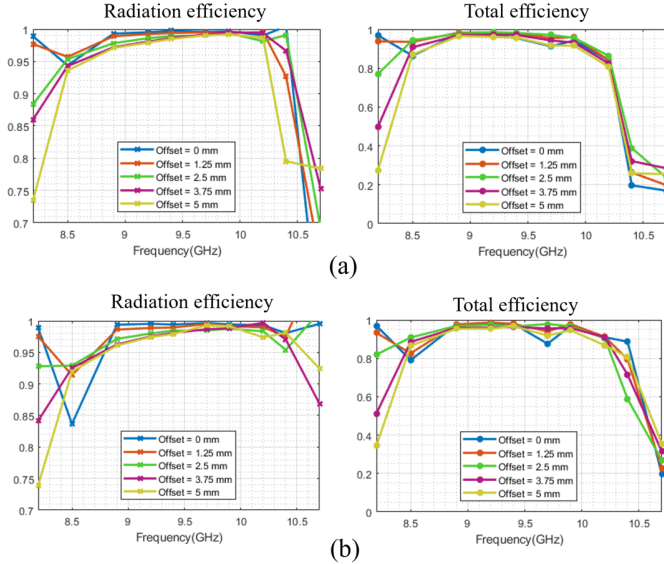


Fig. 20. Radiation and total efficiencies: (a) longitudinal and (b) glide symmetric designs.

B. Radiation Pattern

First, the radiation to the broadside direction (approximately), at a constant frequency of 9.4 GHz is analyzed for different offsets between patches. The beam direction is kept constant in every case as demonstrated in Fig. 21 and Fig. 23. It can also be observed that a larger offset results in a larger antenna gain, what translates to a reduction of the side lobe levels as well.

Regarding the differences between the glide symmetric and longitudinal symmetric schemes, the former presents better side lobe level reduction while the latter presents a larger maximum gain at the cost of a degradation of the side lobe levels.

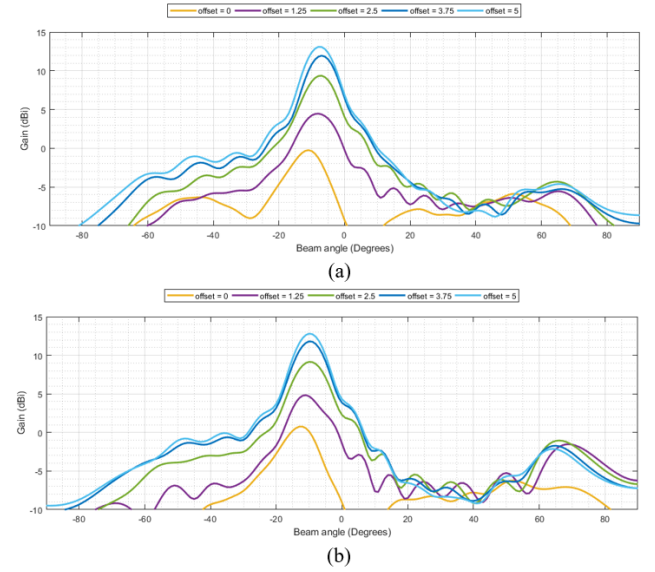


Fig. 21. Cartesian radiation pattern at a constant frequency of 9.4 GHz for (a) the longitudinal symmetric and (b) glide symmetric designs

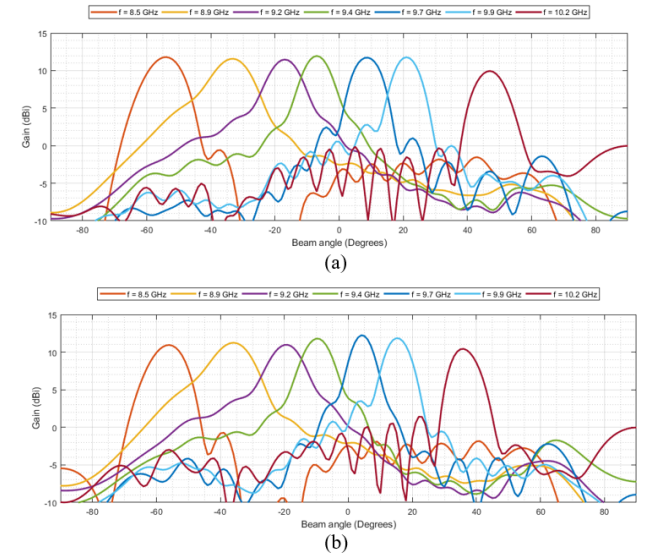


Fig. 22. Cartesian radiation pattern at a constant offset of 3.75 mm for (a) the longitudinal symmetric and (b) glide symmetric designs

In Fig. 22 and Fig. 24 the offset is kept constant at a value of 3.75 mm while the frequency varies along the operational range. As expected, the beam scans with frequency from -47° to 54° , presenting a gain variation of 1.6 dB in the longitudinal symmetric scheme and from -36° to 56° presenting a gain variation of 1.7 dB in the glide symmetric scheme. Therefore, the longitudinal symmetric case achieves a slightly better scanning rate than the glide symmetric version.

V. DISCUSSION

The glide symmetric design for the Goubau line corrugations is recommended for implementation due to its better matching at frequencies above the broadside frequency, as shown in Fig. 19. The recommended offset is 3.75 mm, due

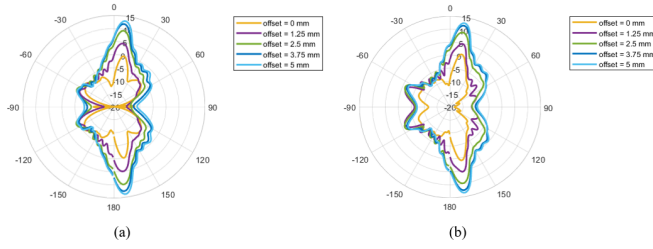


Fig. 23. Polar radiation pattern at a constant frequency of 9.4 GHz for (a) the longitudinal symmetric and (b) glide symmetric designs

to its enhanced open stopband suppression in comparison of the rest of the cases and its high gain.

There are some aspects to be discussed about the final results. Firstly, *CST Microwave Studio Eigenmode Solver* does not support open boundaries, which means that the dispersion diagrams obtained are not exactly the ones expected from the theoretical point of view. Secondly, regarding the radiation patterns, their side lobe levels could be further reduced by tapering the length of the patches along the structure. This would be a way of tuning the attenuation α , which controls the leakage. By using a cosine-shaped illumination this leakage would become more progressive along the length of the antenna, resulting in a higher aperture efficiency. Finally, for a manufacturing stage, cheaper substrates could be used with lower dielectric constants. However, this would require a bigger size of the antenna, as the guided wavelength would be larger and the separation between patches would increase in order to keep a similar performance. If a more compact design is desired, the antenna could be implemented at a higher frequency, as this would reduce the required dimensions for it.

VI. CONCLUSION

In this work a step by step design methodology for an open structure LWA has been reviewed. The main advantages of the presented prototype with respect to other LWA designs are its simplicity for manufacturing and flexible operation. Several parametric studies have been carried out so as to adjust the antenna radiation features to any specific application. We came to the conclusion that both glide symmetric and longitudinal symmetric Goubau line configurations present similar performance when the dispersion curve is shifted from the $\beta p = \pi/2$ to the broadside. The two proposed designs could be used to implement from a low cost fast tracking radar module to an electromagnetic imager.

Future studies can focus on the differences between both schemes when shifting the dispersion curve close to the closed stopband boundary $\beta p = \pi$, as in this region the antenna would present an improved scanning rate. Moreover, the usage of other geometries for the radiation patches (such as circular) and corrugations (such as sinusoidal) could also alter the dispersion characteristics of the Goubau line, featuring a completely different result, which might be of interest for certain applications.

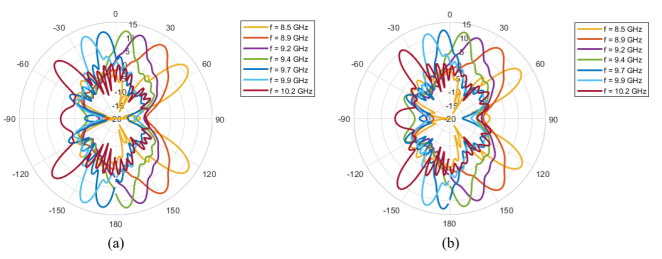


Fig. 24. Polar radiation pattern at a constant offset of 3.75 mm for (a) the longitudinal symmetric and (b) glide symmetric designs

ACKNOWLEDGMENT

The author would like to express special gratitude to the supervisors of the project Shiyi Yang and Qiao Chen for their support, help and advice during the project study. I also would like to thank Oscar Quevedo Teruel for making it possible for me to take part of this project course and Juan-Manuel Rius Casals for supervising the work from my home university.

REFERENCES

- [1] Tang, X.-L. Zhang, and Qingfeng, "Continuous beam steering through broadside using asymmetrically modulated goubau line leaky-wave antennas," *Scientific Reports*, vol. 7, Sep 2017.
- [2] D. R. Jackson and A. A. Oliner, *Antenna handbook. Leaky-wave antennas*. New York: Van Nostrand Reinhold, 1993.
- [3] Q. Li, Y. Zhang, and C.-T. Michael Wu, "Wide-angle frequent scanning metamaterial leaky wave antenna array for automotive radars," in *2018 International Conference on Microwave and Millimeter Wave Technology (ICMMT)*, 2018, pp. 1–3.
- [4] K. Murata, I. Watanabe, A. Kasamatsu, T. Tanaka, and Y. Monnai, "Sparse 3d imaging using terahertz leaky-wave radar," in *2017 42nd International Conference on Infrared, Millimeter, and Terahertz Waves (IRMMW-THz)*, 2017, pp. 1–2.
- [5] C.-H. Tseng and C.-H. Chao, "Noncontact vital-sign radar sensor using metamaterial-based scanning leaky-wave antenna," in *2016 IEEE MTT-S International Microwave Symposium (IMS)*, 2016, pp. 1–3.
- [6] X. Zeng, Q. Chen, O. Zetterstrom, and O. Quevedo-Teruel, "Fully metallic glide-symmetric leaky-wave antenna at ka-band with lens-augmented scanning," *IEEE Transactions on Antennas and Propagation*, pp. 1–1, 2022.
- [7] L. Liu, J. Huang, G. Q. Luo, and Q. Zhang, "Cascaded dispersive delay structure based on periodic glide symmetric microstrip stubs," *IEEE Microwave and Wireless Components Letters*, pp. 1–4, 2022.
- [8] G. Zhang, Q. Zhang, Y. Chen, and R. D. Murch, "High-scanning-rate and wide-angle leaky-wave antennas based on glide-symmetry goubau line," *IEEE Transactions on Antennas and Propagation*, vol. 68, no. 4, pp. 2531–2540, 2020.
- [9] F. Monticone and A. Alù, "Leaky-wave theory, techniques, and applications: From microwaves to visible frequencies," *Proceedings of the IEEE*, vol. 103, no. 5, pp. 793–821, 2015.
- [10] S. F. Mahmoud and Y. M. M. Antar, *Printed Leaky Wave Antennas*. John Wiley & Sons, Ltd, 2010, ch. 13, pp. 435–462. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470973370.ch13>
- [11] M. S. M. I. Mowafak K. Mohsen, "Enhancement bandwidth of half width-microstrip leaky wave antenna using circular slots," *Progress In Electromagnetics Research*, vol. 94, pp. 59–74, 2019.
- [12] Paolo Baccarelli. (2022, Apr). One-dimensional periodic leaky-wave antennas. [Online]. Available: <http://ace1.antennasvce.org>
- [13] I. Wolff, *Coplanar Microwave Integrated Circuits*, 1st ed. Hoboken: Wiley-Interscience, 2006.

Simulation of Microwave Heating of Healthy and Cancerous Human Tissue With Gold Nanoparticles

Hampus Carlens and Mika Söderström

Abstract—With a great need for new and better cancer treatments, microwaves and gold nanoparticles (GNPs) are increasingly being suggested to be used in radiotherapy. In this report, the authors present an investigation of how human tissue, both malignant and healthy, behaves under microwave radiation. Specifically, the focus has been on gradual transitions between healthy and GNP-treated cancerous tissue inside a waveguide. Here, the electromagnetic material property of concern is the relative permittivity. The permittivities of human tissues and GNPs have been modelled using well known analytical models. Furthermore, numerical simulations have been performed using COMSOL Multiphysics and the results have been compared to analytical equations suggested to describe the energy absorption in the material. The study shows that the analytical equations are in agreement with the numerical simulations for lower propagating electromagnetic (EM) modes. Some possible electromagnetic resonance has been seen within the GNP treated cancer tissue. Furthermore, the extensive analytical models and numerical software tools created will be of importance for future research of the feasibility of this cancer treatment method.

Sammanfattning—Med ett stort behov av nya, bättre cancer-behandlingar föreslås mikrovågsstrålning och guldnanopartiklar (GNPs) att användas i strålterapi. I denna rapport presenterar författarna en undersökning av hur mänsklig vävnad beter sig under mikrovågsstrålning. Specifikt har fokus varit på gradvisa övergångar mellan frisk och GNP fylld cancervävnad i en vågledare. Här är den elektromagnetiska egenskapen av intresse den relativa permittiviteten. Permittiviteter av mänsklig vävnad och GNPs har modellerats med välkända analytiska modeller. Vidare har numeriska simuleringar utförts i COMSOL Multiphysics och resultaten av dessa har jämförts med de analytiska ekvationerna som sägs beskriva absorption av energi inuti materialet. Undersökningen visar att de analytiska ekvationerna stämmer överens med de numeriska simuleringarna för lägre propagerande elektromagnetiska (EM) moder. Möjlig elektromagnetisk resonans har setts inom den GNP fyllda cancervävnaden. De omfattande analytiska modellerna och numeriska mjukvaruverktygen som tagits fram kommer vara viktiga för framtida forskning på denna cancerbehandling.

Index Terms—Gold nanoparticle (GNP), Electrophoretic resonance, Dispersive permittivity, Cancer treatment, Waveguide, Radiotherapy, Graded material.

Supervisors: Brage B. Svendsen, Mariana Dalarsson

TRITA number: TRITA-EECS-EX-2022:151

I. INTRODUCTION

With the development of efficient treatment methods for many deadly diseases throughout the last century, cancer may be today's greatest medical challenge. Currently, cancer is often treated using hazardous, ionizing radiation, that unfortunately also harms healthy cells. However, scientists

are increasingly studying the possibility to use microwave radiation instead, in order to reduce the adverse effects of the radiotherapy. This reduction is due to more targeting and non-ionizing radiation.

Breast cancer was the most common form of cancer in the world, regardless of sex, standing for 12.5 % of the newly diagnosed cases in 2020 [1]. For women only, one out of four cancer diagnoses was concerning breast tissue. The domination of breast cancer among all cancer forms constitutes a highly relevant reason to study this tissue. The electromagnetic properties of breast cancer have been the subject of previous experimental studies, see e.g. [2]–[6]. Therefore, it is a qualified tissue to study because of the large amount of data available, compared to other malignant tissues. Moreover, GNPs have already been proved to bind to breast cancer cells in a study made in [7]. In that study, the GNPs serves as a contrast for breast tumour imaging, but the method indicates promising potential to use as a combination of tumour diagnostics and hyperthermia treatment.

A promising cancer treatment method is suggested to use gold nanoparticles covered with ligands. These GNPs are to be injected into the tumour area. Their ligand is specially designed to stick to the cancer cells when absorbed by them. The ligand does not bind well to healthy cells. Cancer cells intake large amounts of nutrients which makes for an even higher concentration of GNPs [8]. Upon binding the GNPs, the cancer is to be radiated with EM waves in the microwave spectrum. Microwaves are electromagnetic waves approximately in the frequency range from 3 GHz to 300 GHz [9]. The applied field will cause the GNPs to oscillate and destroy the cancer cells through internal, local heating [8]. It is important that no significant amount of energy is absorbed by the surrounding tissue in order to not harm it. This suggested treatment is based on theory of electrophoretic (plasmonic) resonance in a lossy background media [10]. To achieve this successfully, one needs to study the behavior of human tissue under microwave radiation.

The use of GNPs in a biological material have previously been studied in a spherical geometry in a spatially unlimited system [8], [11]. An alternative situation is presented in [10], where the system is isolated to a waveguide in order to have more control over the environment. Additionally, a waveguide is suitable for experimental measurements in the future. In previous work [10], a media with smooth material transitions inserted in a waveguide is presented. This type of material, with a graded layer, is more realistic to describe the natural transitions between GNP-treated cancer cells in a tumor and surrounding healthy cells. Furthermore, the graded material

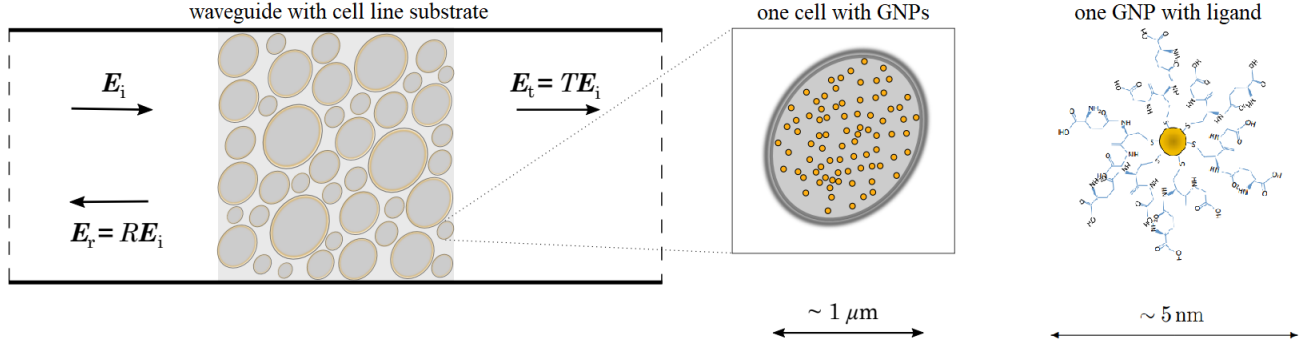


Fig. 1. The physical problem setup, showing a cell line substrate of cancer cells with GNPs inside a waveguide. The illustrations are taken from [10], [14].

can be described by one effective permittivity function which only requires one solution of Maxwell's equations [12].

This project is a continuation of previous work done in [10], where exact analytical equations describing the absorption in a graded material, inserted in a waveguide, were derived. In [13], those analytical equations were confirmed to correspond with numerical simulations, but with arbitrarily chosen constant values for the permittivities. The overall goal of this project was to perform an analytical and numerical study of wave propagation in human tissue, both healthy and GNP-treated cancer tissue, studied in a waveguide. This was done by implementing more realistic models of the frequency dependent permittivity of human tissue, both healthy and cancerous that contains GNPs. The tissues studied in this thesis are breast fat and breast cancer tissue. An overview of the physical problem is presented in Fig. 1. Furthermore, the authors aim to verify the plausibility of using dispersive media in the exact analytical equations derived in [10]. The research presented in this thesis can be of importance towards future studies in this novel field within medicine.

This report is divided into seven sections. After the introduction, some relevant theory is presented, followed by a section about the method of research used, explaining the process in order to obtain the results in the subsequent section. The report is finished with some conclusions and suggestions for future work.

II. NOTATION AND CONVENTIONS

The following notation and conventions will be used throughout the report. The electric and magnetic fields are denoted \mathbf{E} and \mathbf{H} respectively. Time convention $e^{j\omega t}$ is used for all time harmonic fields. Therefore, the complex permittivity of a passive dielectric material will be on the form $\varepsilon = \varepsilon_r \varepsilon_0 = \varepsilon' - j\varepsilon''$, where ε_r is the relative permittivity. The relative permittivity will have a positive real part $\varepsilon' > 0$ and a negative imaginary part $\varepsilon'' < 0$. The real and imaginary part of any complex number ξ is denoted $\text{Re}\{\xi\}$ and $\text{Im}\{\xi\}$, respectively. The angular frequency is denoted $\omega = 2\pi f$ where f is the frequency. The symbols μ_0 , ε_0 and c denotes the permeability, permittivity and speed of light in vacuum, respectively. The free space wave number is defined as $k = \omega\sqrt{\mu_0\varepsilon_0}$. The

acronym GNPs will be used to describe the gold nanoparticles with their ligands attached. In this thesis the term permittivity will always refer to the permittivity relative to that of free space, ε_0 , denoted ε_r .

III. THEORY

The theory section will describe the fundamental theory of wave propagation in rectangular waveguides with inserted materials. The propagation of the waves are not only dependent upon the dimension of the waveguide, but also the properties of the inserted material. The material property of concern in this article is the electromagnetic permittivity $\varepsilon(\omega)$ and the media is considered non-magnetic, i.e the relative permeability is $\mu_r = 1$. Additionally, the electromagnetic properties of biological tissues and GNPs will be explained.

A. Waveguides and electrical field components

A waveguide is a structure in which EM waves can propagate while confined by the boundaries of the waveguide. Waveguide structures can have different shapes and forms [15]. In this thesis the cross-section of the waveguide is rectangular. This type of waveguide is known as a rectangular waveguide, and an example of such is presented in Fig. 2. The cross-section is set to the xy -plane, with wave propagation in the z -direction. The length of the sides of the waveguide in the x and y direction are here denoted d_x and d_y , respectively.

In this structure, ideally, only certain shapes of EM waves exist. More specifically, only shapes that satisfy Maxwell's equations [12] with the given boundary conditions. The boundary conditions in an ideal rectangular waveguide, with perfect electrical conductor walls, are $\mathbf{E} \cdot \hat{\mathbf{n}} = 0$ and $\hat{\mathbf{n}} \cdot \nabla \mathbf{H} = 0$, where $\hat{\mathbf{n}}$ is the unit normal vector of the walls. The different shapes of the EM waves are called modes. In rectangular waveguides only two different types of modes can propagate. Those are the transverse electric (TE) and the transverse magnetic (TM) modes. For the TE modes, the electric field is perpendicular to the direction of the wave propagation, and the analogous is true for the the magnetic field of the TM modes. Furthermore, the modes are defined by the non-negative integers m and n as TE_{mn} and TM_{mn} . For instance, TE_{10} means that the wave only has an electric field component in the y -direction [15].

B. Waveguide cutoff frequency

Wave propagation in the z -direction in a rectangular waveguide is defined by the wave number k_z being [15]

$$k_z = \sqrt{\frac{\omega^2}{c^2} \mu_r \varepsilon_r(\omega) - k_t^2} \quad (1)$$

where the transverse wave number k_t is given by [15]

$$k_t^2 = \left(\frac{m\pi}{d_x}\right)^2 + \left(\frac{n\pi}{d_y}\right)^2. \quad (2)$$

An EM wave is only able to propagate inside a waveguide if $k_z > 0$. This occurs when the operating frequency ω is higher than the cutoff frequency ω_c . The cutoff frequency is the solution to (1) when $k_z = 0$ [15]. The cutoff frequency is dependent on the modes via the integers m and n as well as the dimensions of the waveguide cross-section and inserted material. The mode with the lowest cutoff frequency is called the dominant mode, and is the only mode that will propagate in the structure until the frequency is higher than the cutoff frequency for the next, higher mode. It is common to only let the dominant mode propagate in the waveguide [9]. In the case when $d_x \geq d_y$, TE₁₀ is the dominant mode and the TM mode with the lowest cutoff frequency is TM₁₁.

If the medium is dependent on ω , i.e. dispersive, it implies that the cutoff frequency itself is dependent on the operation frequency. Therefore, one needs to solve the implicit equation

$$\omega_c^2 \mu_r \varepsilon_r(\omega) = c^2 k_t^2 \quad (3)$$

to obtain the cutoff frequency in a dispersive media [9].

C. Reflection and transmission at a material boundary

If the material inside the waveguide consists of two media with different properties, the incident EM wave can either be reflected or transmitted at the boundary between the mediums. The reflection coefficient ρ and transmission coefficient τ denotes the ratio between the incoming and reflected wave, and incoming and transmitted wave, respectively. If the medium is lossy the wave will attenuate, i.e. energy is absorbed by the medium. The Maxwell's equations can be analytically solved inside the geometry of the waveguide by separation of variables, allowing for derivation of the exact analytical equations of power reflection, transmission and absorption, to be described in section III-E.

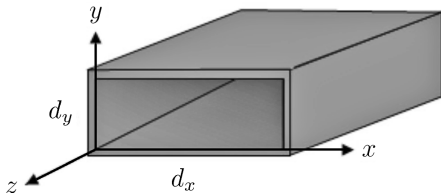


Fig. 2. Illustration of a rectangular waveguide.

D. Graded material in waveguides

Waveguides may contain media with different permittivities. Conventionally, this is described in literature with discrete interfaces between the different media. In [10], a case of a gradual change of material properties is presented. It can be modelled as a single continuous medium with a spatially dependent complex permittivity $\varepsilon_R(\omega, z)$. This can be mathematically described as

$$\varepsilon_R(\omega, z) = \varepsilon_L(\omega) - [\varepsilon_L(\omega) - \varepsilon_G(\omega)] \tanh^2\left(\frac{z}{z_0}\right) \quad (4)$$

where $\varepsilon_G(\omega)$ and $\varepsilon_L(\omega)$ are the complex relative permittivities of the surrounding medium and of the thin layer, respectively [10]. The thin layer is inserted around $z = 0$ with a thickness of $2z_0$. The geometry of graded transitions of materials described by (4) are illustrated in Fig. 3. This is quite a general problem setup, relevant for any waveguide application with smooth material transition into a thin center region inside a waveguide.

E. Power transmission, reflection and absorption

The complex reflection and transmission coefficients, τ and ρ , inside a waveguide with given dimensions and materials can be derived through solving Maxwell's equations. The derivation of ρ and τ over a graded layer described by the permittivity function (4), have been performed in [10], found as equation (39). In a lossless medium the power reflection coefficient and power transmission coefficient are then readily obtained as $R_P = |\rho|^2$ and $T_P = |\tau|^2$, respectively, where $\{R_P, T_P\} \leq 1$ are positive and real valued [10].

In a medium with losses the EM waves attenuate. The electromagnetic energy loss, i.e. energy absorption, is denoted C_{abs} ($0 \leq C_{\text{abs}} \leq 1$) and given by [10]

$$C_{\text{abs}} = 1 - R_P - T_P. \quad (5)$$

Because the media is lossy, there will be absorption throughout the entire waveguide. Furthermore, it is of interest to study the losses in the thin layer only, and therefore exclude the losses in surrounding medium. To compensate for the absorption in the surrounding medium, one needs to add an exact scale factor to the power reflection and transmission coefficients. The exact analytical equations [10]

$$R_P = |\rho|^2 (1 + e^2)^{-4 \text{Re}\{p\}} \chi(a, b, c) \quad (6)$$

$$T_P = |\tau|^2 (1 + e^2)^{-4 \text{Re}\{p\}} \chi(a, b, c) \quad (7)$$

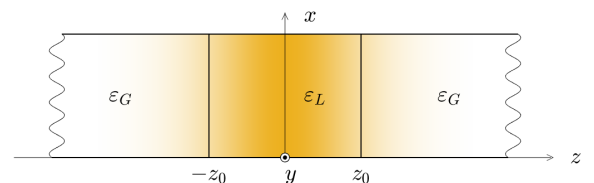


Fig. 3. The geometry of the studied problem. The colors are illustrating the gradient transitions of the materials from ε_G to ε_L , and back to ε_G .

are used to calculate the power reflection and transmission over the thin layer ($-z_0 \leq z \leq z_0$). Here, $\chi(a, b, c)$ denotes an auxiliary scale factor [14] given by

$$\chi(a, b, c) = \left| \frac{F(a, b, c; \frac{1}{1+e^2})}{F(c-a, c-b, c-a-b+1; \frac{1}{1+e^2})} \right|^2 \quad (8)$$

where $F(a, b, c; u)$ is the Gaussian hypergeometric function. Notations, a , b and c have been introduced and are defined as

$$\begin{aligned} a &= 2p + \frac{1}{2} + \sqrt{r^2 + \frac{1}{4}}, \\ b &= 2p + \frac{1}{2} - \sqrt{r^2 + \frac{1}{4}}, \\ c &= 2p + 1 \end{aligned} \quad (9)$$

with the dimensionless variable p given by

$$p = j \frac{k_{zG} z_0}{2}. \quad (10)$$

The dimensionless variable r is different for the different modes as per

$$r_{TE} = k z_0 \sqrt{\varepsilon_L - \varepsilon_G}, \quad (11)$$

$$r_{TM} = \left(k^2 z_0^2 (\varepsilon_L - \varepsilon_G) \left[1 - \left(\frac{1}{\varepsilon_L} + \frac{1}{\varepsilon_G} \right) \frac{k_t^2}{k^2} \right] \right)^{\frac{1}{2}} \quad (12)$$

The z -component of the wave vector when $z \rightarrow \pm\infty$ is denoted as k_{zG} , given by

$$k_{zG} = \sqrt{k^2 \varepsilon_G - k_t^2}. \quad (13)$$

In [14] the power reflection and transmission over some arbitrary layer, hence the absorption via (5), have been derived as

$$R_{Pl} = |\rho|^2 (1 + e^{2l})^{-4 \operatorname{Re}\{p\}} \chi_l(a, b, c) \quad (14)$$

$$T_{Pl} = |\tau|^2 (1 + e^{2l})^{-4 \operatorname{Re}\{p\}} \chi_l(a, b, c) \quad (15)$$

where the real number l denotes the length ($-l \cdot z_0 \leq z \leq l \cdot z_0$) of the arbitrary layer in units of z_0 . The generalized auxiliary scale factor then becomes [14]

$$\chi_l(a, b, c) = \left| \frac{F(a, b, c; \frac{1}{1+e^{2l}})}{F(c-a, c-b, c-a-b+1; \frac{1}{1+e^{2l}})} \right|^2 \quad (16)$$

With the exact analytical equations (6) and (7) and generalized equations (14) and (15), a relative scale factor is then defined as [14]

$$\frac{T_{Pl}}{T_P} = \frac{R_{Pl}}{R_P} = \left(\frac{1 + e^{2l}}{1 + e^2} \right)^{-4 \operatorname{Re}\{p\}} \frac{\chi_l(a, b, c)}{\chi(a, b, c)} \quad (17)$$

and can be applied to both the power reflection and transmission since the scale factor is equal for both coefficients.

F. Permittivity model of human tissue, the Cole-Cole model

The electromagnetic properties of a dielectric material can be obtained from the complex permittivity ε of the given material. In the case of human tissue it has been demonstrated in [16] that the complex permittivity is dependent on frequency, i.e. dispersive. How human tissues respond to a time varying EM field is characterized by the main relaxation regions α , β and γ for low, medium and high frequencies [16], [17]. The relaxation regions are physically explained by polarization mechanisms, where the mechanisms will be different in the respective regions. The polarization mechanism is dependent on the interaction between the tissue and the applied electromagnetic field on a molecular and cellular level [16]. In the γ region, situated at microwave frequencies studied in this thesis, the polarization mechanism is due to the dipolar relaxation of water [16].

For a biological material each relaxation region is often mathematically described with the Debye expression characterized by a single relaxation time¹ τ [18] (not to be confused with the transmission coefficient introduced earlier in section III-C). Since one relaxation region can be characterized by not only one relaxation time, but a distribution of relaxation times, this broadening needs to be taken into account in the mathematical expression. Thus, K. S Cole and R. H Cole introduced the distribution parameter α as a measure of the broadening of dispersion in the biological material. Since each term describes one relaxation region, a summation of N terms will properly represent the permittivity of a human tissue over a larger range of frequencies [16], [18]. This yields the Cole-Cole model

$$\varepsilon_G(\omega) = \varepsilon_\infty + \sum_N \frac{\Delta \varepsilon_N}{1 + (j\omega \tau_N)^{(1-\alpha_N)}} + \frac{\sigma_i}{j\omega \varepsilon_0} \quad (18)$$

where $\Delta \varepsilon = \varepsilon_s - \varepsilon_\infty$ is the magnitude of the dispersion, ε_s is the static permittivity when $\omega \tau \ll 1$ and ε_∞ the permittivity at field frequencies when $\omega \tau \gg 1$, σ_i is the static ionic conductivity. The parameters for the Cole-Cole equation can be determined by a fitting process based on measured data for the respective tissues, as described in [18].

G. Permittivity of cancer tissue

The electromagnetic properties of a malignant tissue is significantly different from that of a healthy tissue. The change in permittivity is due to the high water content in a tumour, compared to the surrounding tissue. A normal healthy breast mostly consists of fat tissue that has a relatively low water content. In the case of a breast tumor the permittivity can be an order of magnitude higher than the surrounding breast fat [16]. When there is a large difference in permittivity between the tumour and surrounding tissue, there is very likely to be a pronounced difference in heating in the respective tissues, which is fundamental for hyperthermia treatments [16].

¹Relaxation time characterizes the time it takes for a material to return to steady state after being exposed to a voltage step, causing a displacement of charge in the interior of the material [16].

H. Drude model and GNP suspension

The permittivity of suspended charged, ellipsoidal particles can be described by the Drude model. In the most generalized form, the Drude model is based on movements of free electrons in a metal [19]. The Drude model can describe movements of gold nanoparticles, for frequencies below the low THz frequency range, as described in [20]. The suspended ellipsoidal particles in a medium can be modelled with the so-called Drude permittivity [11]

$$\varepsilon_D(\omega) = -\frac{\omega_p^2 \tau_D^2}{1 + \omega^2 \tau_D^2} - j \frac{\omega_p^2 \tau_D}{\omega(1 + \omega^2 \tau_D^2)} \quad (19)$$

where τ_D is the relaxation time and ω_p the plasma frequency. The plasma frequency is given by $\omega_p = \sigma_D/(\varepsilon_0 \tau_D)$, where σ_D is the static conductivity. The permittivity of the thin layer can then be written on the form [11]

$$\varepsilon_L(\omega) = \varepsilon_H(\omega) + \varepsilon_D(\omega) \quad (20)$$

where $\varepsilon_H(\omega)$ is the permittivity of the host medium. The host medium is the medium in which the GNPs are suspended.

For electrophoretic particle movements of GNPs, the corresponding Drude parameters are given by $\sigma_D = \mathcal{N}q^2/\beta$ and $\tau_D = m/\beta$. The number of charged particles per unit volume is given by \mathcal{N} , q is the particle charge, β the friction constant of the host medium and m the mass of one GNP. Here, the number of charged particles per unit volume can be modelled as $\mathcal{N} = \phi/(4\pi R^3/3)$, where ϕ is the volume fraction of GNPs in the thin layer and R the total radius of one GNP including the ligands. The net charge of the particle is given by $q = (3R_{Au} + 0.5R_{Au}^2)e_0 + q_L$ with R_{Au} denoting the radius of the core of the gold nanoparticle only and e_0 the charge of one electron. The net charge of the ligands is described by $q_L = n_L e_0$, and n_L is the net electron count in the ligand shell. Furthermore, the friction constant $\beta = 6\pi\mu_f R$ is dependent of the shear viscosity of the host medium μ_f . The mass of one GNP can be calculated with $m = ((4\pi R_{Au}^3)/3)\rho_{Au} + ((4\pi R^3)/3) - ((4\pi R_{Au}^3)/3)\rho_L$, where ρ_L and ρ_{Au} are the mass density of the ligands and gold respectively [20].

As previously stated, several parameters affect the permittivity function (19) and are dependent on both the properties of the suspended particles, in this case GNPs, as well as the host medium. Some of these parameters are constants and others can be varied within a certain range. Varying these parameters is a way of tuning the Drude model [8], causing changes in the electromagnetic properties of the medium. This can be done in order to obtain optimal electrophoretic resonance, i.e. heating, in the suspended GNPs [8].

IV. METHOD

A. Finite Element method

The numerical method used in this project was the finite element method (FEM). FEM is a numerical method for finding approximate solutions to boundary value problems for partial differential equations (PDEs). The general concept of the method is to discretize the geometry into smaller segments called elements. The collective term of these elements is

known as a mesh. After the discretization, the PDE (in this case the wave equation) is solved in each tiny element with applied boundary conditions. Then the set of local solutions is combined into a system matrix A . Finally the linear system $A\phi = b$ is solved, where b is e.g. the applied field and ϕ is the unknown potential [21]. The FEM simulation software used in this project was COMSOL Multiphysics [22].

B. Permittivity models

Human tissues do not have sharp interfaces between GNP-treated cancer cells and healthy cells, but rather smooth transitions between the healthy and malignant tissues. Therefore (4) was used to model the transitions between the tissues.

1) *Surrounding medium*: The relative permittivity ε_G of the surrounding medium (healthy tissue) was modelled with four terms of the Cole-Cole equation (18). The parameters used, here called the Cole-Cole parameters, were based on measured data of breast fat taken from [23], and presented in Table I.

To verify that the Cole-Cole model was correctly implemented in COMSOL, the permittivity in COMSOL was also modelled with interpolation of real measured data from [24]. The results obtained when using the Cole-Cole model was in great coherence with the ones obtained when using interpolation of measured data. This confirmed the implementation of the Cole-Cole model in the given frequency span and allowed the project group to solely rely on the Cole-Cole model when describing the permittivity of breast fat.

2) *Thin layer*: The permittivity ε_L was modelled using (20). The Drude parameters corresponded to the characteristics of GNPs and the host medium ε_H was cancer tissue modelled with the Cole-Cole equation (18).

The tunable parameters in the Drude permittivity (19), described in section III-H, were based on the parameters used in the physical parameter study of GNPs in a saline solution [8], but adjusted for two reasons. Firstly, the parameters were adjusted in order to account for a host medium consisting of malignant breast tissue instead of a saline solution. Secondly, adjustments were made to obtain local extreme points in

TABLE I
COLE-COLE PARAMETERS, BREAST FAT

Parameter	Value
ε_∞	2.500
$\Delta\varepsilon_1$	3.00
τ_1	17.680 ps
α_1	0.100
$\Delta\varepsilon_2$	15
τ_2	63.660 ns
α_2	0.100
$\Delta\varepsilon_3$	$5.00 \cdot 10^4$
τ_3	454.700 μ s
α_3	0.100
$\Delta\varepsilon_4$	$2.00 \cdot 10^7$
τ_4	13.260 ms
α_4	0.000
σ_i	0.010 S m^{-1}

the absorption spectrum described by (5). The parameter dependent on the host medium is the shear viscosity μ_f . For breast cancer tissue the shear viscosity is assumed to be $\mu_f = 2.4 \text{ Pa s}$, based on experimental measurements conducted in [25]. An analysis of how the tuneable Drude parameters for the GNPs affect the absorption was conducted. This was done through fixing all but one parameter and creating surface plots of (5) through (7) over the respective parameter's possible range and frequency. A compilation of the design parameters and constant parameters used for the Drude permittivity (19) are found in Table II and Table III.

The permittivity of the host medium, malignant breast tissue, was modelled with the Cole-Cole equation (18) with one term. The Cole-Cole parameters are presented in Table IV, and are taken from a study made by [2]. The study included 49 samples of breast cancer tissues consisting of $> 30\%$ malignant tissue, measured in the frequency range from 0.5 to 20 GHz. According to [2] the measured data were in agreement with previously published results. Therefore, the values were considered to properly describe the permittivity of breast cancer tissue.

C. General waveguide setup

A rectangular waveguide was set up. It had a finite length of $L = 17 \text{ cm}$ and a cross section with dimensions $6 \text{ cm} \times 3 \text{ cm}$. The walls of the waveguide were perfect electric conductors. The effective relative permittivity $\varepsilon_R(\omega, z)$ inside the waveguide was described by the spatially and frequency dependent permittivity function (4). The permittivity of the surrounding medium ε_G and thin layer ε_L was modelled as described in

TABLE II
DRUDE, DESIGN PARAMETERS

Parameter	Value
n_L	1000
R	2.5 nm
R_{Au}	0.75 nm
ϕ	$2 \cdot 10^{-3}$

TABLE III
DRUDE, CONSTANT PARAMETERS

Parameter	Value
e_0	$1.6 \cdot 10^{-19} \text{ C}$
ρ_L	1000 kg m^{-3}
ρ_{Au}	19300 kg m^{-3}
μ_f	2.4 N s m^{-2}

TABLE IV
COLE-COLE PARAMETERS, BREAST CANCER TISSUE

Parameter	Value
ε_∞	6.749
$\Delta\varepsilon_H$	50.09
τ_H	10.50 ps
α_H	0.051
σ_H	0.794 S m^{-1}

section IV-B, with the thin layer inserted at $-z_0 \leq z \leq z_0$ with a thickness of $2z_0 = 1 \text{ cm}$.

D. Numerical simulations in COMSOL

Numerical simulations in COMSOL were carried out using the setup of a rectangular waveguide as described in section IV-C and illustrated in Fig. 4a. In order to avoid any reflection or unphysical behavior of the outgoing waves at the ports, each port was backed with a perfectly matching layer (PML). The PML absorbed any waves outside of the ports. The mesh used was of custom design. It was dependent on the real part of ε_G and ε_L , and of a specific number of elements per wavelength and therefore frequency dependent. This was done for more efficient and accurate computations. Ten number of elements per wavelength was used. EM waves were excited from one of the ports, as shown in Fig. 4b. Frequency sweeps over approximately a 3 GHz range starting from the cutoff frequency were performed, with 200 equally spaced evaluation points. The cutoff frequency was calculated as described in section III-B. Different modes were excited, these were the dominant mode TE_{10} , and the higher modes TE_{01} , TE_{20} and TM_{11} . The results of the simulations in COMSOL was given as the power reflection R_{Pl} and transmission T_{Pl} over the finite waveguide of length L . In turn the COMSOL data underwent post-processing, to be described in section IV-E.

E. Comparison and analysis in MATLAB

Comparison of the analytical equations and numerical simulations was done in MATLAB. The analytical equations for the power reflection (6) and transmission (7), described in section III-E, were calculated over the thin layer. The results from the numerical solutions gives the power reflection and transmission over the entire waveguide $-L/2 < z < L/2$.

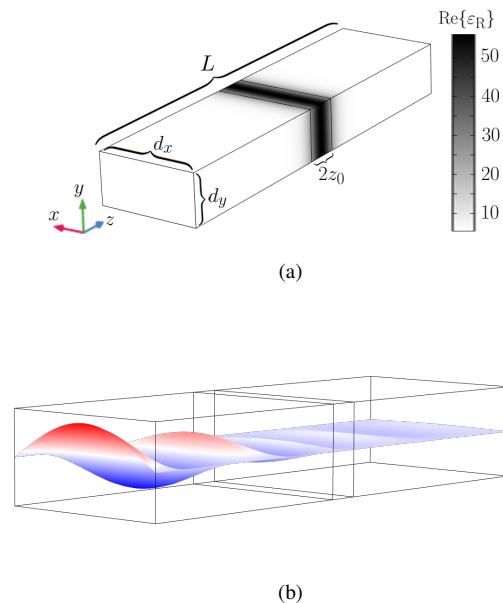


Fig. 4. Waveguide setup in COMSOL excluding the PML boxes, where (a) illustrates the real part of the effective permittivity function of the material and (b) the excited TE_{10} waves at $f = 2.9 \text{ GHz}$ in the finite waveguide.

Therefore, the scale factor (17) needs to be applied with $l = L/(2z_0)$, in order to obtain R_P and T_P over the thin layer only. Additionally, MATLAB was used to calculate and plot; the different permittivity models as well as the error between the numerical simulations and the exact analytical equations.

An extensive MATLAB script was created as a research tool. This script enabled a simple yet powerful way of conducting analysis of the simulations and analytical expressions. The analytical equations were implemented and the script also supported several permittivity models and EM modes. It could generate plots, e.g. comparing the analytical equations with the COMSOL data. Furthermore the script also included simple ways of choosing between its many different functionalities. Another script for studying the influence of the Drude parameters through surface plots was also created. Additionally some useful file formatting scripts written in Python were created to convert the permittivity data into a format supported by COMSOL.

V. RESULTS AND ANALYSIS

A. Permittivities of implemented models

The frequency dependent permittivity of healthy breast fat tissue and GNP treated breast cancer tissue are presented in Fig. 5. The permittivities presented in Fig. 5a of ε_G and ε_L are the minimum and maximum values of the effective permittivity function (4).

There is a significant difference in the magnitude of the permittivity between the surrounding medium ε_G and thin layer ε_L , seen in Fig. 5a. This indicates that hyperthermia can be successful in the case of breast cancer, as previously stated in [16]. However, the permittivity of the GNPs, ε_D , is relatively small compared to the cancer tissue, ε_H , as seen in Fig. 5b. Therefore, the Drude permittivity of the GNPs have a noticeable, but not significant, impact on the permittivity of the thin layer. This insignificant impact could be due to the fact that the movement of the GNPs are much dependent on the shear viscosity μ_f of the host medium. The shear viscosity of breast cancer is high and therefore restricts the movements of the GNPs, possibly reducing electrophoretic resonance. Furthermore, an observed consequence of the high value of μ_f is that a change of the design parameters has a minor effect on the Drude permittivity. This indicates that there is limitations in the permittivity range of ε_D for GNPs suspended in breast cancer tissue.

B. Comparison of simulations and analytical equations for TE_{10}

The power reflection, transmission and absorption over the thin layer are shown in Fig. 6 from both the numerical simulation in COMSOL and the analytical equations (5)-(7). They show an excellent correspondence when comparing the two for the TE_{10} mode. This indicates that the analytical equations formulated in [10] also hold for dispersive media.

The calculated error between the numerical simulation and the exact analytical equation of the power absorption (5) is less than 0.3 %. This possibility to do accurate numerical solutions, even though this problem has an exact analytical solution, is

of importance because future studies may not always allow for the use of the exact analytical expressions. This could be due to, for example, complex geometries and material properties.

Figure 6 shows signs of what could be a local extreme point at around 1.8 GHz, implying possible electrophoretic resonance at these frequencies. Attempting to tune the Drude model for more defined extreme points did not yield any remarkable results. The absorption did not change much when changing the Drude parameters significantly. This also indicates limitations in the GNPs effect on the absorption.

C. Treatment of higher EM modes

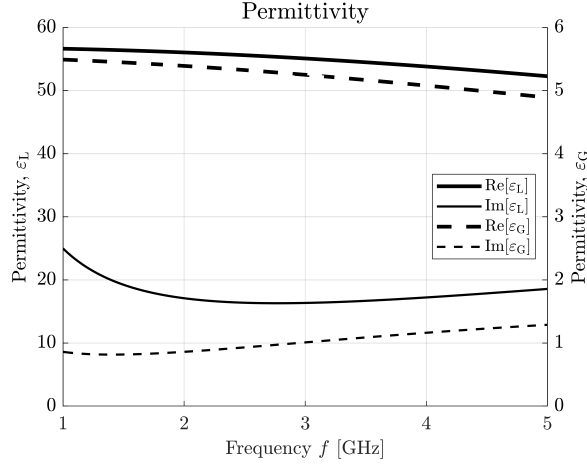
Regarding the higher TE modes TE_{20} and TE_{01} , the COMSOL and analytical results are in as exact agreement as for the TE_{10} mode. As stated in section III-B, it is of interest to only excite the dominant mode in a waveguide. In this thesis the dominant mode is TE_{10} . Therefore figures of results from TE_{20} and TE_{01} are excluded.

Numerical simulation and the exact analytical equations (5)-(7) of the TM_{11} mode are presented in Fig. 7. It is seen that the numerical and analytical results are not in agreement. Oscillations of the numerical results are seen. The origin of these oscillations has not been determined. Furthermore, the numerical results show unphysical behavior, i.e. $C_{abs} < 0$, for frequencies right above cutoff frequency, and at frequencies above 5.5 GHz. It has not been established why this unphysical behavior occurs. It could be due to the implementation of TM modes in COMSOL, the scale factor or something else. However, the unscaled COMSOL data are not apparently unphysical. It is important to note that this has been thoroughly investigated, see Appendix, but no solution has been found and further investigation is needed. The authors therefor suspect that this problem is rather complicated and could be of varying nature.

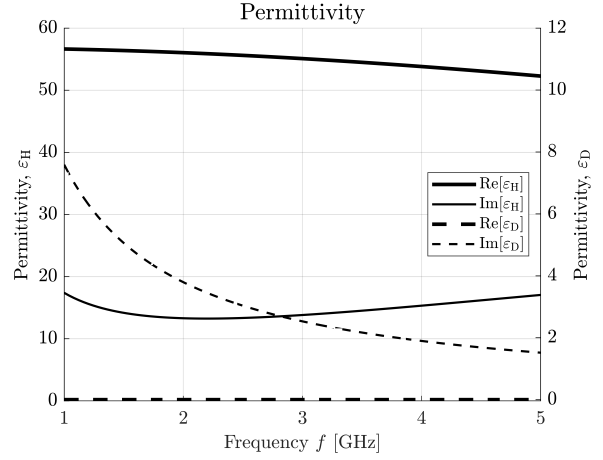
VI. CONCLUSION

In this project, a possible cancer treatment using microwaves to heat up gold nanoparticles (GNPs) inside cancer cells was studied. Specifically, electromagnetic (EM) properties of healthy human tissue were studied, as well as EM properties of GNPs suspended in cancer tissue. This was done by implementing realistic permittivity models of human tissue based on well-known theory, such as the Cole-Cole and Drude model. Additionally, the energy absorption in the GNP-treated cancer tissue inside a waveguide was studied, using graded material transitions between healthy and cancerous tissue. This was studied both with exact analytical equations, derived previously in [10], and by performing numerical simulations using the realistic permittivity models.

The permittivity models for the tissues indicated that hyperthermia can be successful for breast cancer, even though limitations of the contribution by the GNPs were observed. Additionally, it was shown that the simulated power reflection, transmission and absorption for the lower propagating TE modes were in excellent agreement with previously derived exact analytical equations of these entities. This showed feasibility of using realistic dispersive permittivity models in



(a) Permittivity of the thin layer, ε_L , on left axis, and of the surrounding medium, ε_G , on right axis.



(b) Permittivity of the host medium, ε_H , being breast cancer tissue, on left axis, and of the GNPs, ε_D , on the right axis.

Fig. 5. Plot of the real and imaginary part of the frequency dependent permittivity models.

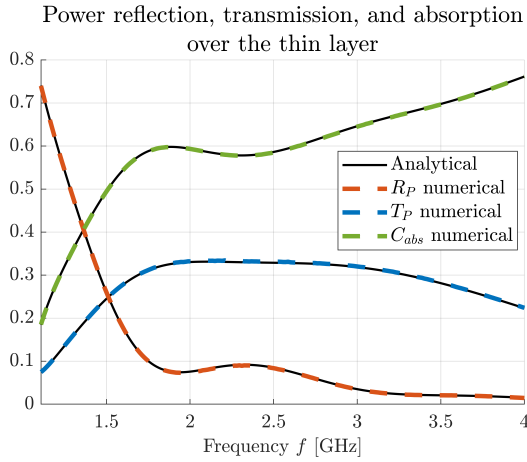


Fig. 6. Exact analytical and numerical simulation results of R_P , T_P and C_{abs} over a range of frequencies, for TE_{10} , with the implemented permittivity models of breast fat tissue and GNP injected cancerous breast tissue.

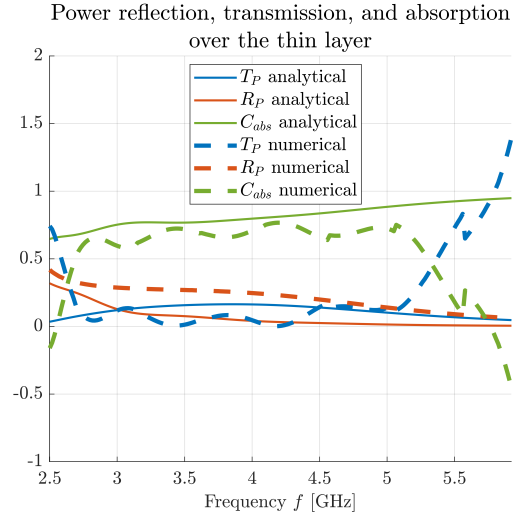


Fig. 7. Exact analytical and numerical simulation results of R_P , T_P and C_{abs} over a range of frequencies, for TM_{11} , with the implemented permittivity models of breast fat tissue and GNP injected cancerous breast tissue.

the exact analytical equations derived in [10]. Furthermore, it was also shown that for the TM_{11} there was discrepancy between the exact analytical expressions and the numerical simulations. The origin of this discrepancy was investigated but not determined.

VII. FUTURE WORK

The conducted study of the permittivity models is of importance for future work on investigating the feasibility of achieving local heating using the described method in the medical application. Future efforts could include a further study on how the Drude model parameters affect the permittivity and if certain choices and combinations could lead to significant electrophoretic resonance of the GNPs at a certain frequency. Additionally, they could include further study on the limitations of the scale factor (17) for the TM mode. Future studies could also aim to develop better simulation models. For example, models that can measure the transmission and

reflection at any part of the waveguide, eliminating the need for using a possibly problematic scale factor.

APPENDIX A

TM & TE MODES, WITH CONSTANT PERMITTIVITIES

APPENDIX B

APPROXIMATE SCALE FACTOR

APPENDIX C

THIN DISCRETE LAYER

ACKNOWLEDGMENT

The authors would like to especially thank their supervisor Brage B. Svendsen for his great guidance, feedback and support throughout the whole project. They would also like to thank supervisor Mariana Dalarsson for her expertise knowledge and support.

REFERENCES

- [1] W. C. R. Fund. (2022, Apr.) Worldwide cancer data. [Online]. Available: <https://www.wcrf.org/dietandcancer/worldwide-cancer-data/>
- [2] M. Lazebnik, D. Popovic, L. McCartney, C. B. Watkins, M. J. Lindstrom, J. Harter, S. Sewall, T. Ogilvie, A. Magliocco, T. M. Breslin, W. Temple, D. Mew, J. H. Booske, M. Okoniewski, and S. C. Hagness, "A large-scale study of the ultrawideband microwave dielectric properties of normal, benign and malignant breast tissues obtained from cancer surgeries," *Physics in Medicine and Biology*, vol. 52, no. 20, pp. 6093–6115, Oct 2007.
- [3] S. Di Meo, P. Espin-Lopez, A. Martellosio, M. Pasian, M. Bozzi, L. Peregrini, A. Mazzanti, F. Svelto, P. Summers, G. Renne, L. Preda, and M. Bellomi, "Experimental validation of the dielectric permittivity of breast cancer tissues up to 50 ghz," in *2017 IEEE MTT-S International Microwave Workshop Series on Advanced Materials and Processes for RF and THz Applications (IMWS-AMP)*, 2017, pp. 1–3.
- [4] S. Takumi, S.-i. Kubota, S.-I. Kuroki, K. Sogo, K. Arihiro, M. Okada, T. Kadoya, M. Hide, M. Oda, and T. Kikkawa, "Complex permittivities of breast tumor tissues obtained from cancer surgeries," *Applied Physics Letters*, vol. 104, pp. 253 702–253 702, Jun. 2014.
- [5] Y. Kuwahara, A. Nozaki, and K. Fujii, "Large scale analysis of complex permittivity of breast cancer in microwave band," *Advances in Breast Cancer Research*, vol. 09, pp. 101–109, Jan. 2020.
- [6] Y. Cheng and M. Fu, "Dielectric properties for non-invasive detection of normal, benign, and malignant breast tissues using microwave theories: Microwave properties of breast tissues," *Thoracic Cancer*, vol. 9, pp. 459–465, Feb. 2018.
- [7] E. Day, "Antibody-conjugated gold-gold sulfide nanoparticles as multifunctional agents for imaging and therapy of breast cancer," *International Journal of Nanomedicine*, vol. 5, p. 445, Jun. 2010.
- [8] S. Nordebo, M. Dalarsson, Y. Ivanenko, D. Sjöberg, and R. Bayford, "On the physical limitations for radio frequency absorption in gold nanoparticle suspensions," *Journal of Physics D: Applied Physics*, vol. 50, no. 15, p. 155401, Mar. 2017.
- [9] D. M. Pozar, *Microwave engineering; 3rd ed.* Hoboken, NJ: Wiley, 2005.
- [10] M. Dalarsson, Y. Ivanenko, and S. Nordebo, "Wave propagation in waveguides with graded plasmonic obstacles," *J. Opt. Soc. Am. B*, vol. 38, no. 1, pp. 104–113, Jan 2021.
- [11] M. Dalarsson, S. Nordebo, D. Sjöberg, and R. Bayford, "Absorption and optimal plasmonic resonances for small ellipsoidal particles in lossy media," *Journal of Physics D: Applied Physics*, vol. 50, no. 34, p. 345401, Jul. 2017.
- [12] J. C. Maxwell, "A dynamical theory of the electromagnetic field," *Phil. Trans. R. Soc.*, pp. 459–512, Jan 1865.
- [13] B. B. Svendsen and M. Dalarsson, "Numerical study of te-wave propagation in waveguides with graded plasmonic obstacles," in *2021 IEEE USNC-URSI Radio Science Meeting (Joint with AP-S Symposium)*, 2021, pp. 90–91.
- [14] B. B. Svendsen, M. Söderström, H. Carlens, and M. Dalarsson, "Analytical and numerical models for te-wave absorption in a graded-index gnp-treated cell substrate inserted in a waveguide," *Submitted to Applied Sciences*, 2022.
- [15] D. K. Cheng, *Field and wave electromagnetics*, 2nd ed. Harlow, Essex, England: Pearson, 2014.
- [16] K. Foster and H. Schwan, "Dielectric properties of tissues and biological materials: A critical review," *Critical reviews in biomedical engineering*, vol. 17, pp. 25–104, 1989.
- [17] C. Gabriel, S. Gabriel, and E. Corthout, "The dielectric properties of biological tissues: I. literature survey," *Phys. Med. Biol.* 41, pp. 2231–2249, 1996.
- [18] C. Gabriel, S. Gabriel, and R. W. Lau, "The dielectric properties of biological tissues: III. parametric models for the dielectric spectrum of tissues," *Phys. Med. Biol.* 41, pp. 2271–2293, 1996.
- [19] W. Cai and V. M. Shalaev, *Optical Metamaterials: Fundamentals and Applications*. New York, NY: Springer, 2010.
- [20] E. Sassaroli, K. C. P. Li, and B. E. O'Neill, "Radio frequency absorption in gold nanoparticle suspensions: a phenomenological study," *Journal of Physics D: Applied Physics*, vol. 45, no. 7, p. 075303, Feb. 2012.
- [21] J.-M. Jin, *Theory and computation of electromagnetic fields*, 2010.
- [22] COMSOL. (2022, Jan.) Comsol multiphysics simulation software. [Online]. Available: <https://www.comsol.com/comsol-multiphysics>
- [23] C. Gabriel and S. Gabriel. (2022, Apr.) Compilation of the dielectric properties of body tissues at rf and microwave frequencies, appendix c: Modelling of the data. Physics Department, King's College London. [Online]. Available: <http://niremf.ifac.cnr.it/docs/DIELECTRIC/AppendixC.html#Sum>
- [24] C. Gabriel and S. Gabriel. (2022, Apr.) Compilation of the dielectric properties of body tissues at rf and microwave frequencies, appendix d: Experimental data. Physics Department, King's College London. [Online]. Available: <http://niremf.ifac.cnr.it/docs/DIELECTRIC/AppendixD.html>
- [25] S. Alikhani, M. A. Ansari, and A. R. Niknam, "Simulation of thermoacoustic resonance response of tumor by finite element method," *Journal of Applied Physics*, vol. 126, no. 17, p. 174701, 2019.

CONTEXT J

FUSION – THE SUN’S ENERGY SOURCE ON EARTH

POPULAR DESCRIPTION

Future Energy Might be Hotter than the Sun

Growth. Creation. Life. None would be possible without the sun. As our society grows, so does our demand for energy. Temperatures are rising as a result of our current energy production, jeopardizing the wellbeing of the planet. Could creating a miniature sun on Earth result in a nearly endless source of clean energy? According to the science of fusion – the answer is yes.

Fusion is a nuclear process where two light elements are combined into one heavier one. The process releases energy and is what powers the sun, and all other stars in the universe. On Earth, fusion requires temperatures as high as 100 million degrees, several times the temperature of the core of the sun. The immense heat brings significant challenges as no material can withstand more than a few thousand degrees without melting. At these temperatures the fuel used for fusion is in the form of a plasma, which can be contained by using some of the most powerful magnetic fields created by humans.

The main fuel-source of fusion reactions is deuterium, which can easily be extracted from seawater. This makes fusion a virtually limitless power source. Similarly to the nuclear power of today, fusion also produces radioactive waste. The amount of waste, however, is much smaller and has a shorter life span. Fusion also has advantages over other renewable sources such as solar and wind power. It is not dependent on the weather and can always produce a lot of energy in a small area.

Climate change is one of the greatest challenges of our generation, fueled by greenhouse gas emissions creating more violent weather than ever before. With 72% of the world's emissions coming from energy production there is big room for improvement. Since fusion power provides clean and emission-free energy, our fossil fuel dependency and climate impact could be greatly reduced for countless future generations.

SUMMARY OF PROJECT RESULTS

Fusion is the process where light elements are fused together into heavier ones. The energy released in this reaction corresponds to the difference in mass between the products and reactants according to $\Delta E = \Delta mc^2$. Due to the large amount of energy produced there is currently a lot of research on how to create fusion-based reactors where the freed energy is used to produce electricity. One common reaction being researched is $D + T \rightarrow He + n + 17.6 \text{ MeV}$, which involves deuterium (D) and tritium (T), two different isotopes of hydrogen. This reaction is one of the most energy efficient and is therefore planned to be used in future fusion power plants.

To achieve a fusion reaction, high temperatures of over 100 million degrees Celsius are required. At these temperatures the fuel is ionized, creating a plasma. The plasma needs to be confined in a way that does not create damage to its surroundings. This can be accomplished in a torus shaped device called a tokamak, where the plasma is confined with magnetic fields. The inner wall placed closest to the plasma has to be made from durable materials to withstand the heat from the plasma and protect components from radiation. Over time, the inner walls have to be replaced due to damage caused by instabilities in the plasma.

In project J1 the group members studied a method used for analyzing old wall-samples from experimental fusion reactors which could provide crucial information on how the plasma interacts with the walls. This data has the potential to be used to make future reactions more efficient. The method called ToF-ERDA works by hitting the sample with a high-energy ion beam which can knock out atoms from the wall material. The energy and speed of the knocked out particles can be detected and

used to create a depth-profile where the concentration of different elements are plotted as a function of the depth. Current methods for analyzing, and simulating ToF-ERDA data assume that the target sample is perfectly flat, which in the real world tends not to be the case. The goal of project J1 was to examine how roughness can affect the results gathered from ToF-ERDA. This was done by writing two different programs, which could apply roughness to output data from two already existing simulation programs: TRIM and Potku. By combining data from multiple simulations in TRIM and Potku respectively, a rough surface could be modeled.

The results of this project will help to create a better understanding of rough samples in TOF-ERDA and in turn help provide a basis for more accurate depth-profiles in the future. This will provide a clearer insight of how wall materials are affected by the conditions inside fusion devices. Better wall materials could therefore be selected and the interactions between plasma and wall could be optimized. Future studies could implement the results into the analysis software.

In project J3 the group members analyzed one method of plasma heating called ion cyclotron resonance heating (ICRH), which utilizes radio waves. Under the right circumstances, the radio waves resonate with the ions of the plasma and power is absorbed. The aim of project J3 was to simulate and compare the heating of plasma between two fusion facilities that vary in size and plasma content: ASDEX Upgrade (AUG) and ITER. AUG is a smaller tokamak located in Germany while ITER is currently being built in France and will be the largest tokamak in the world when it is completed. The focus of this project was to analyze the structure of the propagating radio waves in the plasma. This was achieved by performing relevant simulations of ICRH for different scenarios in AUG and ITER. ITER simulations were made both for a plasma consisting of only deuterium and for a plasma with an equal mix of deuterium and tritium. The simulations of AUG were only made for a pure deuterium plasma.

The results of the simulations showed a clear difference in wave propagation between the two facilities. ITER, which was simulated with a larger plasma radius than AUG, had radio waves behaving more as a beam than as an eigenmode pattern in the plasma. In contrast to this, simulations of AUG showed waves propagating more as an eigenmode pattern than as a beam. In the ITER simulations, the amount of reflected waves was lower and more power was absorbed by the plasma than in AUG.

The obtained results further show that ITER will be more effective than smaller fusion facilities like AUG. The bigger dimensions and different plasma content result in more power absorption that is more localized to the center of the plasma. For future studies, plasma heating through ICRH could be compared to cyclotron resonance heating of electrons (ECRH).

IMPACT ON SOCIETY AND ENVIRONMENT

Climate change is one of the greatest challenges of our generation and carbon dioxide emissions from energy production is one of the main culprits. Fossil fuels currently dominate the sector and most renewable alternatives are intermittent, meaning that their energy production is dependent on external factors such as the amount of wind or sunshine. Fusion is often seen as a possible solution to these problems as it can provide a virtually emission-free reliable base energy source. A problem with this argument, however, is that no functioning fusion power plant is yet to be constructed and there is still uncertainty regarding if, or when, a fusion reactor will be commercially viable. Even if fusion energy alone does not solve the climate crisis, it is still an interesting alternative for future green energy.

A big advantage of fusion power is the availability of fuel. One of the main power sources of fusion is deuterium which can be extracted from sea water. It is therefore a lot cheaper and more abundant than the fuels we use today which means that it could help reduce the dependencies on other fuel rich countries. The production of tritium, the other required isotope of hydrogen, does however consume lithium which means that the world's dependence on lithium exporters may, in turn, rise. To exacerbate the problem, the lithium used in the reaction is turned to helium, meaning that recycling will not be possible.

A common concern with nuclear energy is that it enables the creation of nuclear weapons. This is also the case for fusion energy. Because of the complexity of a fusion reactor however, it is less likely to be used for this purpose. The refinement process needed for fission would be a more straightforward way of obtaining such weapons, but it is important to note that it will be possible with fusion reactors as well.

In contrast to fission power, fusion power does not pose any risk of a nuclear meltdown as the reaction is simply slowed to a halt at a critical failure. Because of this, fusion power plants would not pose the same risk to the surrounding area as fission power plants in case of a natural disaster or a war. Furthermore, nuclear fusion results in considerably less radioactive waste compared to fission. The radioactive isotopes that are produced also have to be contained for around 100 years, compared to 100 000 years for the isotopes of fission power. This reduces the risk of information about the confinement being lost or distorted over time. Overall, fusion power is a safer source of energy for society compared to nuclear fission. This could be an important argument for fusion to tackle the trend of skepticism that exists towards the nuclear power of today.

The society is dependent on reliable and safe energy sources that can provide energy at all times. While fusion would be a good source regarding this, it could also contribute to a more centralized energy system. Because of the high energy-density in fusion reactors, society could get more dependent on fewer sources. This could in turn create major power outages in case of operational disturbances or catastrophes. An advantage of high energy-density though, is that it leads to more efficient use of land than for example solar or wind power that needs very big areas to produce a sufficient amount of energy. Fusion power is in that way impacting the environment to a lesser extent.

Electricity has proven to be an essential part of enabling sustainable development and an increased production could potentially increase the quality of life for billions of people. The centralization and large scale of individual reactors, however, means that they require a large upfront cost and a developed electric grid. Large parts of the world simply lack the electric grid or money required and fusion will therefore not be of any use in the immediate future. Only once larger grids have been established and the economic situation has improved, fusion can be considered a viable option.

If it turns out that fusion reactors can be used commercially, it may offer a preferable source of power compared to what is available today. This could affect other industries in the energy sector, where fossil fuels are used. There is already a lot of money and resources invested in fossil fuel based power plants and it is therefore not economically feasible to shut these plants down once they are up and running. Power plants that utilize fossil fuels may therefore still be operating, even after more sustainable alternatives have been created. This could result in a slower transition to sustainable energy sources, which is bad for the environment. On the other hand, a slower transition will most likely result in fewer job losses within the sector of fossil fuels.

Accelerator-Based Analysis of Rough Wall Materials From Fusion Devices

Fabian Persson Djurhed and Vilhelm Forkman

Abstract—Time of Flight - Elastic recoil detection analysis (ToF-ERDA) is a method used to analyse the composition of wall samples from fusion devices. All current analysing software for ToF-ERDA assumes that the target is perfectly flat which could create inaccuracies when rough surfaces are analysed. The aim of this project was to get a better understanding of how the roughness of samples from fusion devices affect the results from ToF-ERDA. To investigate this, three existing simulation software SIMNRA, TRIM and Potku were used. Programs were developed in order to use these to simulate three different targets with varying roughness, which were modelled as a combination of surfaces of different thicknesses. The results from which were put back into Potku where the differences between the targets could be noted. The study shows that it was possible to apply roughness to the already existing programs and showed similarities between the resulting depth profiles. When applying roughness, the concentration of surface elements decreased at the top of the layer but also went further into the sample.

Sammanfattning—Time of Flight - Elastic recoil detection analysis (ToF-ERDA) är en metod som används för att analysera kompositionen av prover av väggmaterial från fusionsreaktorer. Alla mjukvaror som används för att analysera datan från ToF-ERDA idag antar att provets yta är helt platt vilket skulle kunna innebära att felaktiga resultat erhålls när så ej är fallet. Målet med där här projektet var att undersöka och skapa en bättre förståelse för hur skrovligheten hos material från fusionanordningar påverkar resultaten från ToF-ERDA. För att undersöka detta användes tre simulationsmjukvaror, SIMNRA, TRIM och Potku. Program skrevs för att använda dessa för att simulera tre olika material med olika stor skrovlighet, vilka modellerades som en kombination av material med olika tjockt ytskikt. Resultaten från dessa analyserades därefter i Potku där skillnaderna mellan materialen kunde noteras. Studien visar att det är möjligt att implementera ojämnheter i den simulationsprogram som finns idag och flera likheter mellan de resulterande djupprofilerna från de olika simulationsmetoderna uppmärksammades. När högre skrovlighet användes minskade koncentrationerna av ytelementen vid materialets topp men djupet som de når i materialet ökade.

Index Terms—Potku, TRIM, SIMNRA, ToF-ERDA, Roughness, Simulations.

Supervisors: Laura Dittrich, Per Petersson

TRITA number: TRITA-EECS-EX-2022:152

I. INTRODUCTION

Material research for fusion devices has been an important topic over the last 50 years of fusion development [1]. In fusion devices, wall materials have to withstand heat and neutron irradiation emitted from the plasma. A substantial amount of the energy produced in fusion reactions is the kinetic energy in neutrons. This kinetic energy translates into

high velocities that can cause damage and erosion when the wall materials absorb the energy. Particles can also reflect back into the plasma causing impurities that will later be deposited elsewhere on the wall materials [2].

To understand how wall materials are affected by the harsh environment inside fusion devices they have to be analysed. One method for this is called Time of Flight - Elastic Recoil Detection Analysis (ToF-ERDA) and can detect all elements present on the surface of a sample. The data from ToF-ERDA is then often used to make depth profiles that show how the atomic concentration of elements changes with depth [3]. The data collected from ToF-ERDA can be crucial in figuring out how the plasma interacts with the walls of the reactor which in turn enables the development of better walls and more efficient future reactors.

According to [4] depth profiles for rough materials with ToF-ERDA show an incorrect representation of depth because of assumptions that the sample surfaces are flat. In this project we aim to examine how roughness of wall materials affect the depth profiles obtained with ToF-ERDA. To understand how depth profiles of rough surfaces could be improved we intend to explore how simulations of rough surfaces could be made from three programs TRIM, Potku and SIMNRA.

Prior work to this project has been the development of the three simulation-programs mentioned above. Even though these programs and analysing methods have been improved, applying roughness to simulations is still not possible in TRIM and Potku. One of the programs, SIMNRA, has implemented some methods to apply roughness for simulations according to [5].

II. THEORY

A. ToF-ERDA

Elastic recoil detection analysis (ERDA) is an ion beam analysis technique that is non-destructive for sample materials [6]. The method consists of an ion beam of up to 100 MeV striking a sample also called target when used in analysis. Some atoms of the target get knocked out after being hit and are then called recoils. There is also something called multiple scattering and is when a collision does not immediately create a recoil. Instead of knocking out an atom this will start a chain of multiple internal collisions in the target before a recoil eventually leaves the target. ERDA is based on the detection of these elastically recoiled atoms and collects the energies [3]. With only energy data it can be hard to discriminate the detected atoms, therefore there is another method that also measurements time of flight called time of flight ERDA (ToF-ERDA). As seen in Fig. 1 the ion beam hits the target at an

angle α knocking atoms with angle θ from the beam path hitting an energy detector. To measure the time of flight (ToF) the recoils go through two carbon foils that start and stop the time measuring [4]. ToF-ERDA is therefore the combination of measuring both time and energy for all atoms hitting the energy detector.

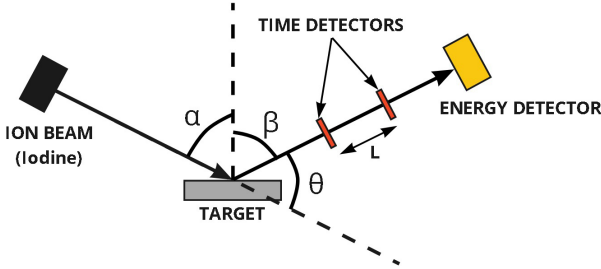


Fig. 1: Setup of ToF-ERDA for our project.

B. SIMNRA

SIMNRA was once one of the most widely used simulation programs for ToF-ERDA and was popularised by the upgrades after its predecessors. With simulations running both quicker and with a more forgiving user interface than earlier programs [7]. SIMNRA is an analytical simulation program that bases its calculations on mathematical models and approximations. This means that it can run simulations in just a matter of seconds compared to programs based on Monte Carlo simulations that have a substantially longer run time. Roughness in SIMNRA is calculated through a combination of different target thicknesses. These can be combined in some different ways in SIMNRA, from a user input distribution file, a gaussian distribution or a gamma distribution. Because of this it's possible to simulate arbitrary rough surfaces of different targets [5].

C. TRIM

TRIM, which stands for “the Transport of Ions in Matter”, is a program that was first developed by James F. Ziegler in 1985 and has since been continuously updated, with the latest update being released in 2013. According to [8] the program uses the Monte Carlo method to calculate the path and energy loss of individual particles as they travel through a material. The calculations, showcased in [9], are based on what is known as “The magic formula” which was first developed by J. P. Biersack. The formula allows for relatively fast computing times while maintaining a high accuracy. According to Ziegler’s result presented in [8] the average discrepancy of the simulated particle compared to experimental data is 4.3%. There is currently no option to simulate rough targets in TRIM. In this project TRIM is used as a physics engine to simulate different particles’ range and motion in the target material, the data from which is used to model an ToF-ERDA setup.

D. Potku

Potku is a software primarily designed for ToF-ERDA and can be used to make depth profiles, which are graphs that show

the atomic concentration of the elements at different depths in the target [10]. Potku is based on the input of experimental data from laboratories and can then be used to visualise and analyse the data. In order to analyse, data about counts, energies and ToF are needed. From this data the depth profiles can be made which is one of the tools that Potku has.

Another useful function in Potku is the ability to do Monte Carlo simulations of ToF-ERDA and has settings that make it possible to mimic real setups [11]. Simulation results can then be compared with real data from similar setups. The simulations in Potku are based on the code described in [12] which also say that it is partly built on similar code to TRIM which is described earlier. Potku’s simulations can be used to make energy spectras that give all the relevant data to be able to create depth profiles.

III. METHODOLOGY

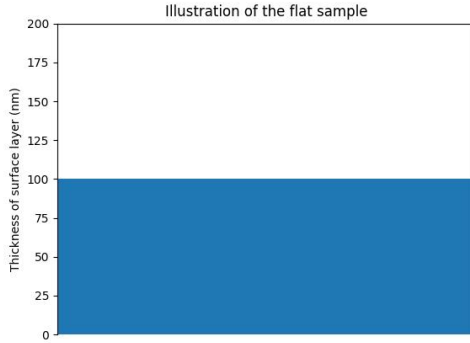
A. Target and experimental setup

The experimental setup being simulated in the different methods was chosen to mimic the Tandem laboratory in Uppsala which is illustrated in [4]. A 36 MeV iodine beam at an angle of $\alpha = 67.5^\circ$ was used in all simulations as well as a detector angle of $\beta = 67.5^\circ$. 400 mm was chosen as the distance between ToF detector one and two in accordance with [4].

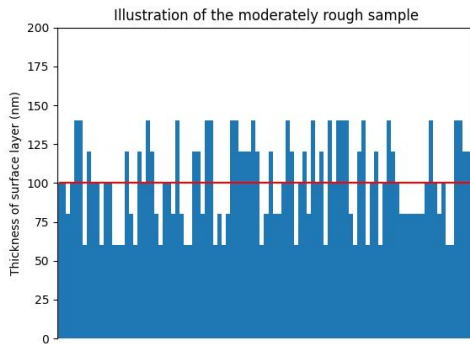
Since this project was fusion based the samples or “targets” were chosen to be relevant for this. Inside modern fusion devices elements such as beryllium and tungsten are commonly used [1]. This made beryllium and tungsten particularly interesting to use for our project. As described earlier the wall materials get contaminated with other elements inside the fusion devices over time, which is why our target is not only beryllium and tungsten. To make the target more realistic, other elements such as carbon and oxygen which often are present in real samples were added [1]. To be specific, our target consisted of two layers, a surface and a substrate. The substrate layer consisted of 100% tungsten and the surface layer an equal atomic concentration of beryllium, carbon, oxygen and tungsten. Apart from beryllium and tungsten that are used in fusion devices, oxygen and carbon were added to the surface mixture to represent a more realistic target that has some deposited elements on top.

Three different surface roughnesses were used all of which had an average surface thickness of 100 nm. The first target which is illustrated in Fig. 2a was completely flat with a uniform thickness, which functioned as the control. The two remaining targets represented a moderate and an extreme case. Both of them were modelled as being a combination of five different possible thicknesses. The moderate case had a maximum deviation of 40% from the average with thicknesses of 60, 80, 100, 120 and 140 nm which is illustrated in Fig. 2b. The extreme case had a maximum deviation of 80% from the average with thicknesses of 20, 60, 100, 140 and 180 nm which is illustrated in Fig. 2c. The substrate was made 500 nm thick to make sure the beam would not pass through. The units for the input data varied between nm in Potku and TRIM to $1e15 \text{ at./cm}^2$ in SIMNRA. The conversion rates between the two

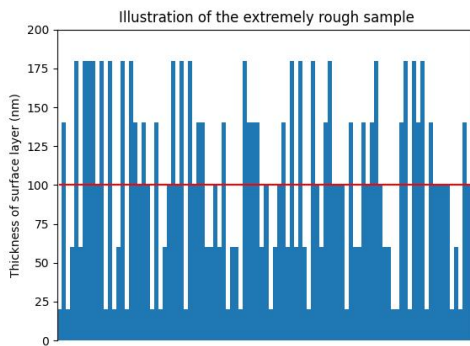
units depend on the composition of the material. To make the conversion between these the density calculator in SIMNRA was used together with TRIM that could calculate the density of the surface layer. For the flat target, the cutoff point between surface and substrate of 100 nm roughly corresponds to a thickness of $800 \text{ } 1\text{e}15 \text{ } \text{at.}/\text{cm}^2$.



(a) Flat surface



(b) Moderate roughness (60 - 140 nm)



(c) Extreme roughness (20 - 180 nm)

Fig. 2: Visualisation of the surface roughness implemented in this project.

B. Calculating ToF

Simulations from TRIM, SIMNRA and Potku result in energy spectras. This data is not directly of any use since Potku requires data in the form of energy, ToF and counts. The ToF could be calculated with the formula for kinetic energy

$$E = \frac{mv^2}{2} \quad (1)$$

where m is the mass of the recoil. According to this formula the fastest particle, beryllium with a kinetic energy of 36 MeV, has a velocity of $2.7 \cdot 10^7 \text{ m/s}$ which corresponds to around 9% of the speed of light. It was therefore decided that relativistic effects would not be large enough to warrant being accounted for. With the knowledge of the distance L as shown in Fig. 1, a formula for the ToF was derived as

$$t = \frac{L}{\sqrt{\frac{2E}{m}}} \quad (2)$$

Equation 2 together with energy-data from simulations gave all the needed values to calculate the ToF for all energies.

C. TRIM

Both SIMNRA and Potku simulations are designed to mimic a real setup and will therefore provide data on the energy, mass and counts of all elements that are simulated to hit the detector. TRIM on the other hand is designed to simulate an ion's range and path when it travels through a material. For every ion going into the material, it therefore only provides information about the energy and angles of the ion leaving the material. In order to use this information to simulate an ToF-ERDA measurement, a program was created in Python 3.10 using the integrated development environment PyCharm that we called "RunTRIMinPython".

Collisions that create recoils are rare events and would take a long time to gather the necessary data without doing any sort of modification to TRIM. It was therefore decided to simulate a number of ions going to different "recoil points" at evenly spaced depths throughout the material and then manually calculate the recoil directions, cross-sections and energy being transferred in the recoil. In order to do so the program manipulates the text file TRIM.DAT which makes it possible to specify different starting positions, angles and energies for all the ingoing ions. TRIM only provides data of ions leaving the material so it was necessary to use a negative angle, meaning that the simulated path is actually that of an ion going towards the surface from the recoil points instead of the other way around. Since the material an ion has to travel through to reach the surface from a certain depth is the same as the material going in the opposite direction, they are equivalent and the data can be interpreted as the same.

For flat surfaces only one simulation was needed for the ions going into the target and to their respective recoil points. After the calculation of the recoil data, which is covered in section III-D, one simulation was needed for each element in the target in order to simulate their outwards paths. The fact that the outwards and inwards simulations were done separately made it possible to run multiple outwards simulations for every one inward simulation. When considering the rough targets, the outwards paths were therefore simulated using all the five different surface thicknesses for every one inwards simulation. This allowed the recoils to exit the target through a different amount of material than they entered, which is a better model

for targets with a rapidly changing surface thickness. For five inwards simulations, 25 times more outwards simulations had to be run, compared with the flat surface. In order to keep the total amount of simulated ions constant for all three targets the rough targets used 25 times fewer ions per simulation, resulting in more spread-out recoil points.

D. Calculation of recoil and cross-sections

The energy that was transferred to the recoil in the collision was calculated using equation 3 specified in [13].

$$E_2 = E_0 \frac{4m_1m_2}{m_1 + m_2} \cos^2\theta \quad (3)$$

Where E_2 is the energy being transferred to the recoil, m_1 and m_2 are the masses of the incoming ion and the recoil respectively. θ is the angle between the directions of the incoming ion and the recoiled particle.

Cross-sections can be interpreted as the area which needs to be hit for a specific collision to happen. A larger cross-section would mean that that specific collision is more likely to happen than a collision with a smaller cross-section. According to [13], the standard Rutherford recoil cross-section can be calculated by using

$$\sigma_R^{ERD} [mb/sr] = 2.0731 \cdot 10^7 \frac{\{Z_1 Z_2 (m_1 + m_2)\}^2}{(2m_2 E [keV])^2 \cos^3\theta} \quad (4)$$

Z_1 and Z_2 are the atomic numbers of incoming ion and the recoil respectively. The Rutherford cross-section is, however, slightly inaccurate. It's been experimentally found that the Rutherford cross-sections are not completely accurate for both high and low energies. For recoil angles $\theta < 90^\circ$ the Andersen correction

$$F_{Andersen} = \frac{(1 + \frac{1}{2} \frac{V_1}{E_{CM}})^2}{(1 + \frac{V_1}{E_{CM}} + \{2E_{CM} \sin^2 \frac{\theta_{CM}}{2}\})^2} \quad (5)$$

has been developed and should be multiplied with the Rutherford cross-section to get better results. The equation for $F_{Andersen}$ uses a centre of mass coordinate system, as opposed to the lab based reference frame of all previous calculations. The angle and energies used therefore needed to be

$$\theta_{CM} = \theta + \arcsin\left(\frac{m_2}{m_1} \sin\theta\right) \quad (6)$$

and

$$E_{CM} = \frac{m_2}{m_1} E_R \quad (7)$$

respectively, where E_R is the energy of the incoming particle in the laboratory system. V_1 is the increase in kinetic energy of the system and was calculated using

$$V_1 [keV] = 0.04873 Z_1 Z_2 \sqrt{Z_1^{2/3} + Z_2^{2/3}} \quad (8)$$

The cross-sections were calculated for every element and recoil angle in every recoil point and multiplied with the concentration of the given element at the specified depth. The product was finally divided by 10000 and rounded to the

nearest integer. The resulting "scale factor" was interpreted as the probability of a recoil happening as compared to all other recoils. Or more precisely how many data points every recoil should represent. The number 10000 is completely arbitrary but it was found to strike a good balance between getting a reasonable number of data points and not excluding too many low probability recoils.

E. Angles

As suggested by Eq. 4 particles with higher energies have a lower chance of interacting with the material it is passing through. This is not just true for events resulting in recoils but for other types of interactions as well. As demonstrated by Fig. 3, low energy particles are therefore more likely to have their paths curved. In a real world laboratory setup there is a wide array of possible recoil angles. Some recoils whose initial direction after impact is straight towards the detector might curve off in the target and miss. Other recoils that were travelling in the wrong direction initially might instead curve back towards the detector as a result of these interaction. Ideally the program should work in the same way by simulating particles recoiling in all directions. This would however require the simulation of many more particles since most of them would not hit the detector. It was therefore decided to simulate five different initial recoil directions for each recoil point. The direction was defined as being in a random angle between zero and five degrees on either side of the direction of the detector in the plane spanned by the initial direction of the ion going into the target and the direction of the detector. The energy and scale factors was calculated for every one of these angles and for every element in each recoil point using the methods described in section III-D

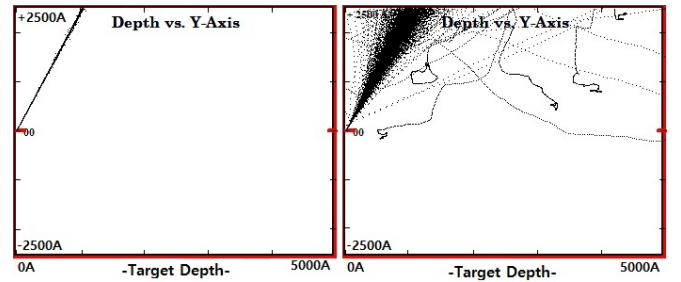


Fig. 3: The path of 1000 carbon atoms with 36 MeV (left) and 2 MeV (right) through a 500 nm thick tungsten layer with an initial angle of $\alpha = 67.5^\circ$ being simulated in TRIM. The y-axis parallel to the target surface in the direction of the detector and the x-axis is the depth into the target. Every dotted line corresponds to the path of one particle.

The limited range of the recoil angles does reduce computing time but combined with the small detector size it also favours the detection of high energy particles since there are no wide angle recoils to curve back towards the detector. To compensate for this, seven different detector sizes were used. The detector sizes correspond to 7 different angle ranges with centre in the direction of the detector. If a particle's direction, when exiting the material, is within one of these angle ranges then it was considered to have hit the detector. The first angle

that was considered is $\arctan(1/100)^\circ$, or around 0.57° . This corresponds to an angle that is not off by more than one centimeter from the center of the detector for every one meter travelled, which is similar to the real world setup. The six larger angle ranges considered were 1° , 2° , 2.5° , 5° , 10° , 90° all of which corresponds to detectors of increasing size. Ideally only the first angle, $\arctan(1/100)^\circ$, would need to be considered and a larger variety in the different recoil angles would be simulated. The only downside to doing so is time, which is why the recoil angles were restricted and all the different detector sizes were used.

F. Energy loss and output files

The energies and directions of the recoils were put back into TRIM where their energy loss and path out of the material and towards the detector was simulated. The time of flight values of these particles were calculated using Eq. 2. In order for the units to fit within the 0 - 8000 range that is required for Potku, all energy values were converted to MeV and multiplied by 200 and all time of flight values were converted to nanoseconds and multiplied by 15. When writing the final data to a file the number of events were determined by the scale factor and noise was added following a normal distribution with a standard deviation of 15 to both the energy and time of flight values, in their respective units, in order to get more realistic data. Seven different output files were generated, corresponding to the seven different sizes of detector.

G. Simulations in Potku

Simulations were made in Potku with settings that mimic the behaviour of a real ToF-ERDA set up. In our case simulations were run using the same setup as the Tandem Laboratory in Uppsala with measurement settings as described in III-A and detector settings shown in Appendix A. From the simulations an energy spectra of the simulated elements could be created. When the energy spectra was created, Potku automatically saved the data in files for each element individually. To be able to make depth profiles out of this, the data had to be processed in Matlab.

H. Processing data from Potku simulations

Output data from Potku is made up from a text file for each and every element in the simulated target. The text file contains two columns with data of energy in MeV and amount of observed atoms (counts) that hit the detector. From this data it is not directly viable to make a depth profile though. To create a depth profile in Potku, energy and ToF is needed as input. The ToF had to be calculated from the energy and mass of each element as shown in Eq. 2. A Matlab program was then written to be able to take in data from energy spectras, calculate the ToF and then write a text file with the energy and ToF. We chose to name this Matlab program "RoughPotku" for easier referencing and can be seen in Appendix D.

By only using this file as input to Potku, information about the number of counts was lost. To include the counts in the two

column text file, every count was made into separate events. This was done by printing multiple lines to the output file corresponding to the number of counts in every point. By doing this Potku could get information about energy, ToF and counts in the two columns of data that was needed as input.

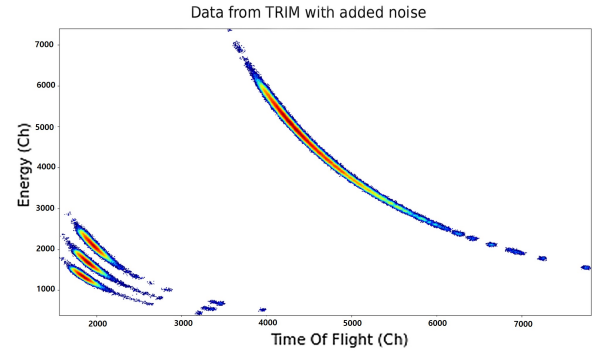


Fig. 4: Produced bananas after adding noise from TRIM simulations.

The graph that shows the ToF, energy and counts in Potku has gotten its "banana" shape shown in Fig. 4 from the square dependency between energy and ToF shown in Eq. 2. Since the ToF was calculated for every value of energy and then copied, we had multiple equal values of ToF for every energy. Because of this, the bananas that were created were perfectly shaped but without any thickness. From real data the bananas have a thickness that makes it possible to see how the counts are distributed. To make our bananas look more natural some noise was implemented to the energy and ToF-data. The noise was created by altering the copies of ToF and energy according to a normal distribution. The mean was set to the original value of the data with a standard deviation of 15. With the implementation of noise the bananas looked more realistic.

To be able to apply the roughness aspect on the data from Potku, multiple simulations were run with different surface thicknesses which is similar to the method that SIMNRA uses. Five simulations were made in Potku with thicknesses ranging from 60 - 140 nm in equal step size for the moderate case. The same procedure was made for the extreme case with thicknesses from 20 - 180 nm. In RoughPotku the data could be combined from the five simulations into one file by stacking all data for each element. For example all energy and count data from beryllium was stacked to make one beryllium dataset. After the energyspectras had been combined RoughPotku turned the data into the bananas that Potku could make depth profiles out of, as explained above.

As described in section III-F the data had to be scaled to fit the range in Potku. The scaling for RoughPotku was made to be the same as in TRIM to get similar values.

I. Simulations in SIMNRA

The simulations were run with settings corresponding to the setup explained in section III-A. Since roughness could be applied directly in SIMNRA from a roughness distribution file it was applied using a textfile consisting of two columns, surface thickness and frequency. The input files is shown in

TABLE I
MODERATE ROUGHNESS INPUT FILE FOR SIMNRA

Thickness [$1e15 \text{ at./cm}^2$]	Frequency
0	0
479.9	0
480	1
480.1	0
639.9	0
640	1
640.1	0
799.9	0
800	1
800.1	0
959.9	0
960	1
960.1	0
1119.9	0
1120	1
1120.1	0

TABLE II
EXTREME ROUGHNESS INPUT FILE FOR SIMNRA

Thickness [$1e15 \text{ at./cm}^2$]	Frequency
0	0
159.9	0
160	1
160.1	0
479.9	0
480	1
480.1	0
799.9	0
800	1
800.1	0
1119.9	0
1120	1
1120.1	0
1439.9	0
1440	1
1440.1	0

Table I and II and had the same roughness distribution as used in Potku and TRIM. SIMNRA interpreted this file as if the probability of a certain thickness was the same for the whole surface. The thicknesses in the tables will therefore be weighted the same when simulating in SIMNRA using this file, similar to what has been done with RoughPotku. In both TRIM and Potku multiple scattering was simulated, but in SIMNRA though, multiple scattering had to be turned on to account for these events as explained in section II-A.

J. Processing data from SIMNRA simulations

The data from simulations that were run in SIMNRA needed to be processed in a similar way as Potku and can be seen in our Matlab-program "SIMNRAImport" in Appendix E. In SIMNRA the simulations result in an energy spectra as in Potku, but the main difference is the output data. In SIMNRA all the data could be written from the spectra to one file. This meant that the file contained information about all the elements and even isotopes from the energy spectra which was useful when writing the code. In the file masses for each isotope could be found, which made calculating ToF with Eq. 2 a bit easier than from Potku data. The scaling for SIMNRA was the same as in RoughPotku and RunTRIMinPython.

K. Calibrating data

To be able to make a depth profile from the simulation-data it had to be calibrated, which is done in Potku. Because detectors of different setups act differently, a calibration is needed both when processing real data and when processing simulated data to go from "channel numbers" of ToF and energy to proper units. Since the data from our simulations were processed a bit differently in the three programs written, calibrations were needed for all three. A calibration in Potku needs at least three different elements to be accurate, which is because the calibration is based on a linear regression between the elements. With our target having four elements at the surface as described in section III-A calibration could be made directly from the data of our codes. But to get an even more

accurate calibration a "calibration-file" was also made where six elements, beryllium, carbon, oxygen, silicon, titanium and iron, were simulated to get more data for the linear regression. Lighter elements were used as heavier ones can show some strange characteristics which are unwanted in the calibrations.

IV. RESULTS

In this section the resulting depth profiles made from the combined energyspectras in RunTRIMinPython, SIMNRAImport and RoughTRIM is shown together with flat simulations for comparison.

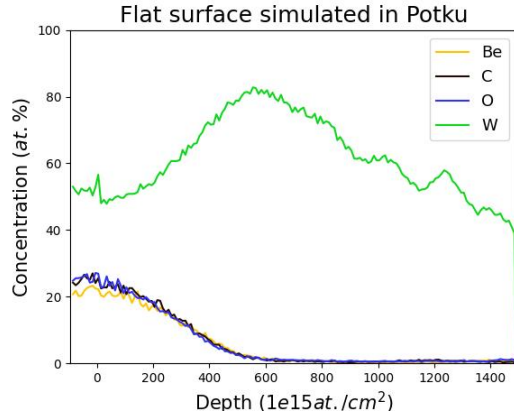
A. Depth profiles from Potku

Fig. 5 show the different roughnesses simulated in Potku. When comparing the depth profiles, there are some patterns that could be pointed out. Firstly the tungsten has a peak that gets more flattened out with more roughness. Secondly, the concentration of the surface elements is higher closer to the surface, which corresponds to the left side of the graph, in the rough target. The concentration at the surface for beryllium, carbon and oxygen decreases slightly with more roughness. This difference is seen most clearly when comparing the flat surface with the extreme case, where it decreases from an atomic concentration of roughly 25% down to about 21%. Note that when concentrations are mentioned, it is the atomic concentration of the elements that is discussed and not the mass concentration. Thirdly an increase in how deep the surface elements reach into the target can be seen where the concentration approaches 0%. The flat surface shows that the surface elements reach a depth of about $600 \text{ } 1e15 \text{ at./cm}^2$, while the moderate target reaches about $700 \text{ } 1e15 \text{ at./cm}^2$ and the extreme case reaches over $800 \text{ } 1e15 \text{ at./cm}^2$.

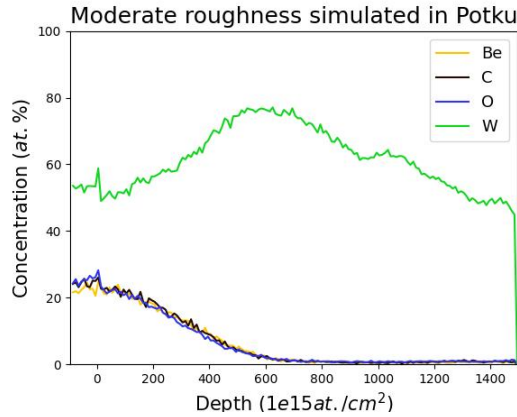
B. Depth profiles from SIMNRA

In SIMNRA three roughness simulations were made in Fig. 6. When simulating the moderate and extreme case, Table I and Table II were used as input roughness files. When comparing the extreme case to the flat case the peak has

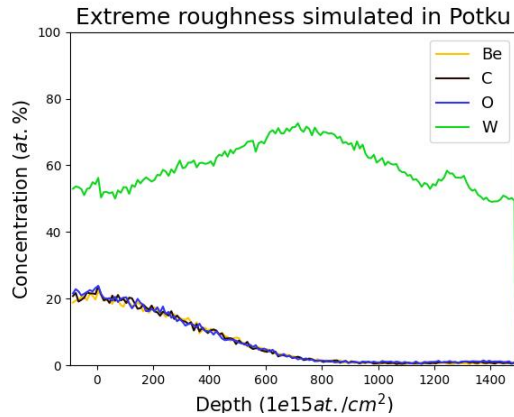
been almost completely flattened out for tungsten up to about $800 \text{ } 1\text{e}15 \text{ at./cm}^2$ where it starts to slope down. The starting concentrations for the surface elements are about 30% for the flat and moderate case, the extreme case shows a slightly lower concentration. The start and ending concentration of tungsten is about constant for all cases where it starts at roughly 52% and ends at 23%.



(a) Flat case

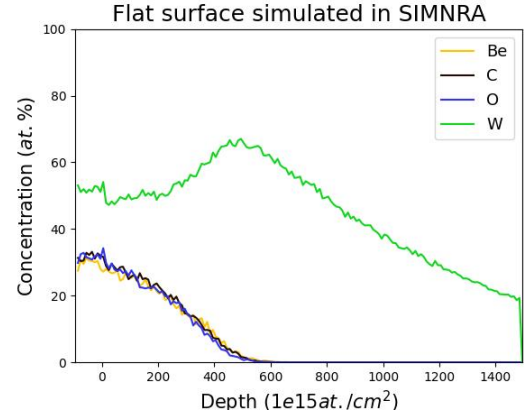


(b) Moderate Case

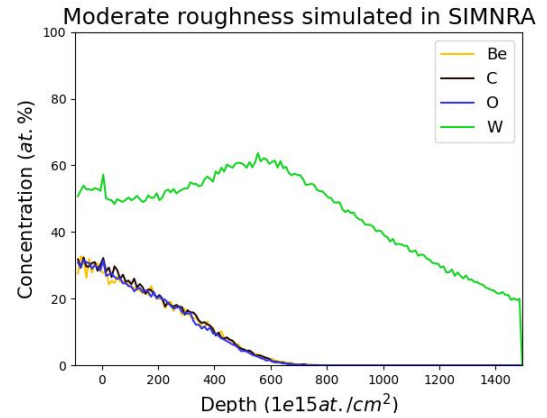


(c) Extreme case

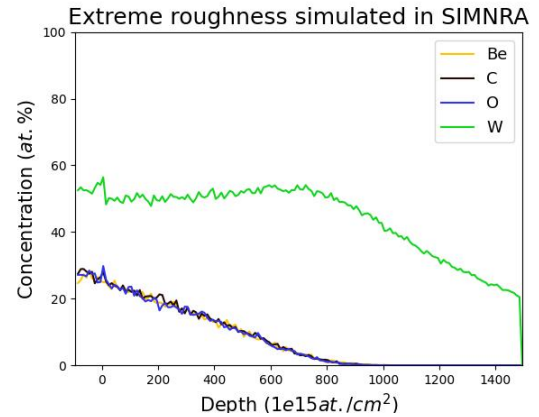
Fig. 5: Depth profile of Flat, moderate and extreme roughness simulated in Potku.



(a) Flat case



(b) Moderate Case



(c) Extreme case

Fig. 6: Depth profile of Flat, moderate and extreme roughness simulated in SIMNRA.

C. The effects of roughness in TRIM

When taking all three targets into consideration, see Fig. 7, there is little to no difference between the flat target and the target with moderate roughness that can not be explained by random variations between simulations. When compared to the other two, the extreme roughness appears to have lower concentrations of beryllium, oxygen and carbon. Tungsten on the other hand seems to have a higher concentration throughout the target in the rough cases as compared to the flat case. In

Fig. 7 there seems to be no noticeable difference in how deep the surface elements reach into the target. For other angle spans, all of which are showcased in Appendix C, the surface appears to reach slightly deeper into the target in the case of increased roughness.

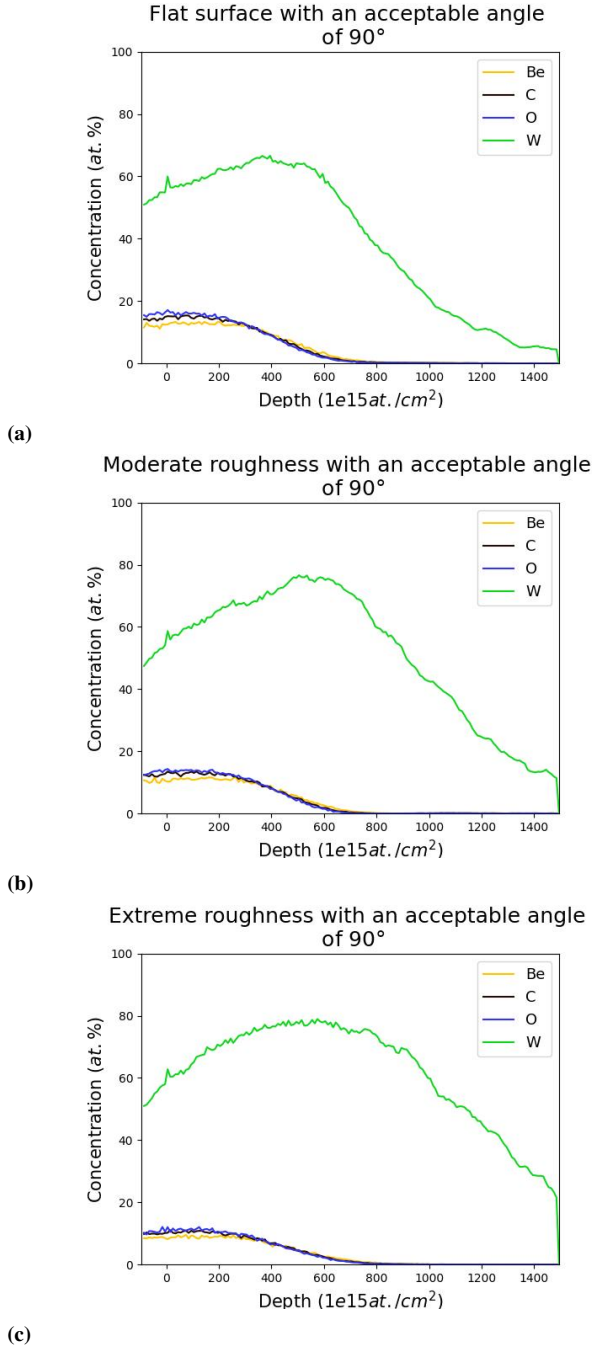


Fig. 7: The three different targets simulated in TRIM with an acceptable angle of 90°.

D. Comparison of results

The results from Potku and SIMNRA were virtually identical, see Fig. 5 and Fig. 6. In both cases the surface elements had lower concentrations for higher roughnesses but reached deeper into the target. Both programs also showcased very similar tungsten curves, with a more pronounced peak in the

flat target which flattens out as the roughness increases. The concentration of tungsten decreases faster after the peak in SIMNRA than Potku however. In TRIM the results were more subtle. The concentrations of the target elements seemed to decrease for rougher surfaces but there was not a big difference in how far the surface reach into the different targets. Due to how much the tungsten curves change for the different acceptable angles shown in Fig. 8 and 9, no further results can be gathered from the shape of the tungsten curves.

V. DISCUSSION

A. Evaluation of RunTRIMinPython

Judging from Fig. 8 and 9 it became clear that the acceptable angle plays a big roll in determining what the depth profile will look like. The main difference is that the concentration of tungsten is increased throughout the sample when more generous acceptable angles are considered. In this particular project this is not a large issue since the goal of the project was only to examine the effects of roughness. All comparisons in the results are done between different targets that have been simulated using the same acceptable angle. It was therefore possible to draw conclusions about how roughness affects the results in any one of the different acceptable angle setups separately. The results that were similar across most, or all, cases were then assumed to be the general results for how roughness affected the depth profiles.

In order to make RunTRIMinPython accurately simulate any rough material more studies need to be done, particularly in the selection of both the acceptable angle and the recoil angles. In our research, the results from the 90° acceptable angle, see Fig. 9d, is the one that is the most similar to the results gathered from Potku and SIMNRA. The Potku simulation software uses a method, detailed in [12] which assigns a statistical weight to all recoils based on how far from the centre of the detector they hit. Similar methods could be implemented in the RunTRIMinPython program.

B. Multiple scattering

When our simulated results are compared with what they theoretically should look like as Fig. 10 shows, a lot of differences can be seen. One clear thing is that tungsten slopes down instead of going to 100% when the surface elements have decreased to 0%. This could partially be explained by the fact that simulations include multiple scattering. Multiple scattering is when a recoil interacts with multiple other particles on its way out of the target. When simulations were made in SIMNRA with the multiple scattering turned off, the tungsten increased to roughly 100% after the depth passed the surface layer as shown in Fig. 11. Multiple scattering is automatically included in the simulations conducted in TRIM and Potku. For heavy elements in particular, the effects from multiple scattering can not be ignored. The results indicate that Potku, as an analysis software, does not consider the effects of multiple scattering when creating the depth profiles. When analysing real world data from tungsten samples these effects could heavily influence the results. To get more accurate results for heavy elements such as tungsten, the ion beam energy

could be increased in future studies to see if the downward slope-effect decreases. Since multiple scattering affects the depth resolution according to [14] it could be the factor making the graph slope downwards. For our results we therefore look more at the point where the tungsten concentration increases as a sign of where it would go to 100%. This also has to do with the fact that Potku normalises the data between 200 - 400 $1e15 \text{ at./cm}^2$, which is why that range gives the most valuable information.

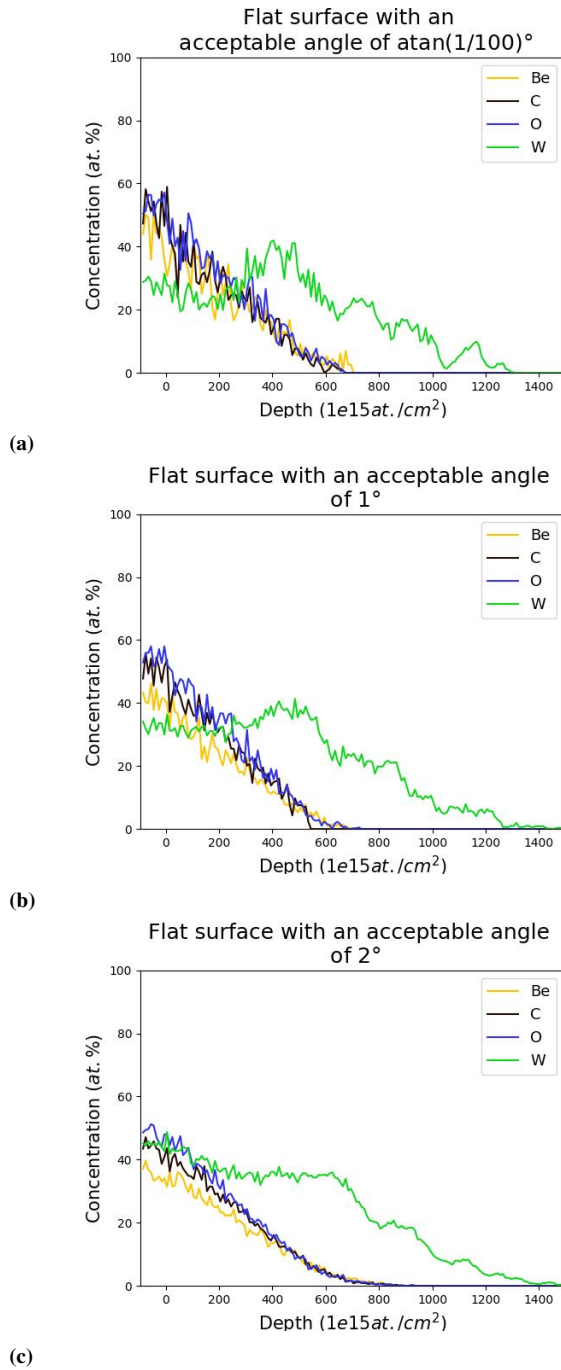


Fig. 8: Acceptable angle span for flat surfaces simulated in TRIM ranging from $\text{arctan}(1/100)^\circ$ to 2° .

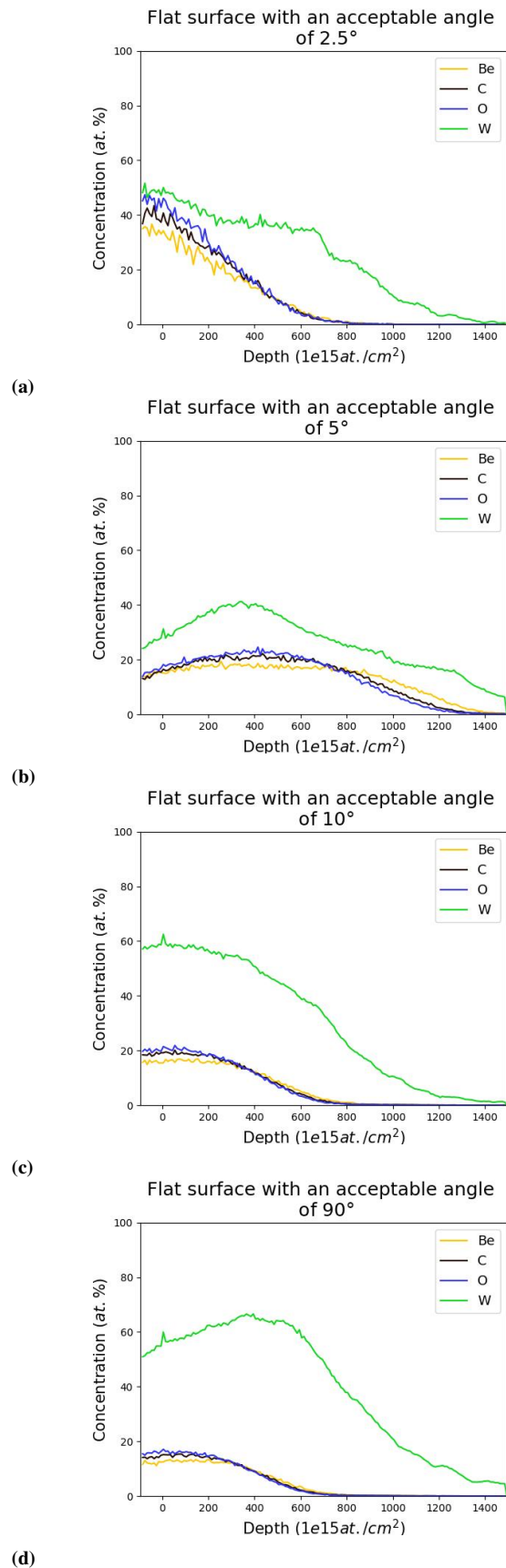


Fig. 9: Acceptable angle span for flat surfaces simulated in TRIM ranging from 2.5° to 90° .

C. Concentration difference between elements at the surface

Worth noting is the large difference between the concentration of tungsten and the other elements at the surface. Since this was true for all simulations in SIMNRA, Potku and the large acceptable angles of TRIM, see for example Fig. 9d. It appears as if this is a problem with how Potku calculates the concentrations for the depth profiles rather than a problem with any one simulation software. There are more counts of tungsten overall since the substrate only has that one element which might influence Potku's calculations for the surface. Further studies are needed to determine why this large deviation from the theoretical depth profile in Fig. 10 occurs and by how much it affects the results of real targets.

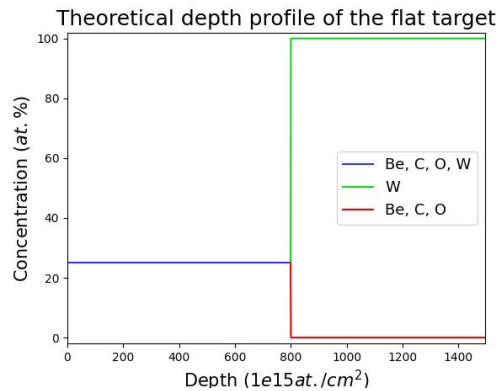


Fig. 10: Theoretical depth profile of flat target.

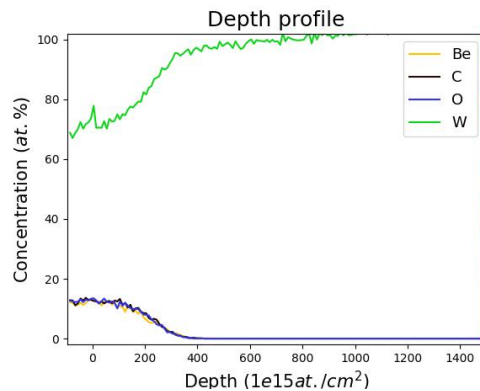


Fig. 11: Flat surface simulated in SIMNRA without multiple scattering.

D. The effects of roughness

In Potku and SIMNRA simulations the surface elements reached further into the target for more rough materials. This is to be expected since the way roughness is modelled in this project is by keeping the average thickness constant while adding and removing an equal amount of material above and below to get the desired roughness, which is illustrated in Fig. 2. The maximum thickness of the surface in the rough targets is therefore larger than the 100 nm of the flat target, meaning that the surface should reach deeper in the depth profiles. In the TRIM simulations this effect was not as apparent. This could be due to the difference in how the

rough targets were modelled in TRIM compared to the other two methods. In Potku and SIMNRA five separate simulations were, corresponding to the five different thickness levels. In TRIM however, the outwards path was simulated using all five thickness levels for every inwards path that was simulated. Only 1/25 of all recoils were simulated at the thickest level for both the inwards and outwards path, compared to 1/5 of all recoils in Potku and SIMNRA. Since the average thickness is constant the material of the surface is more spread out, meaning that there will be fewer counts for each depth. While, in reality, the concentration are constant, Potku will interpret the decrease in counts as a lower concentration which is also shown in the results.

VI. CONCLUSION

The programs written in this project have shown that applying roughness to simulations from existing programs is possible. The results from TRIM varied a lot depending on the accepted angle but still showed some similarities to SIMNRA and Potku. When a rough surface was applied the concentration of the surface elements decreased at the top of the surface. Roughness also made the surface elements reach deeper into the target which was expected.

The effects of multiple scattering were bigger than the effects of roughness in our case. It would therefore be interesting to investigate further how multiple scattering affects in Potku and TRIM simulations could be reduced to make roughness effects more visible. The effects from multiple scattering, and possibly the increased surface concentration, become the most apparent for heavy elements. In fusion research tungsten is a common element to use for the walls of reactors. It is therefore especially important that researchers that use ToF-ERDA to analyse wall samples from these reactors are aware of and compensate for these effects. Simulations of rough materials should be conducted more often to determine whether the results acquired from Potku are accurate or whether they are a just result of how Potku handles heavy elements.

APPENDIX A DETECTOR-SETTINGS IN POTKU

APPENDIX B PYTHON CODE FOR RUNTRIMINPYTHON

APPENDIX C ALL DEPTH PROFILES FROM TRIM

APPENDIX D MATLAB CODE FOR ROUGHPOTKU

APPENDIX E MATLAB CODE FOR SIMNRAIMPORT

ACKNOWLEDGMENT

The authors would like to thank their supervisors Laura Dittrich and Per Petersson for their great support, helpfulness and availability all throughout the project which has been greatly appreciated.

REFERENCES

- [1] J. Linke, J. Du, T. Loewenhoff, G. Pintsuk, B. Spilker, I. Steudel, and M. Wirtz, "Challenges for plasma-facing components in nuclear fusion," *Matter and Radiation at Extremes*, vol. 4, no. 5, p. 056201, Aug. 2019. [Online]. Available: <https://doi.org/10.1063/1.5090100>
- [2] M. Rubel, "Fusion neutrons: Tritium breeding and impact on wall materials and components of diagnostic systems," *Journal of Fusion Energy*, vol. 38, no. 3, pp. 315–329, Aug. 2019. [Online]. Available: <https://doi.org/10.1007/s10894-018-0182-1>
- [3] W. Assmann, H. Huber, C. Steinhausen, M. Dobler, H. Glückler, and A. Weidinger, "Elastic recoil detection analysis with heavy ions," *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, vol. 89, no. 1, pp. 131–139, 1993. [Online]. Available: [https://doi.org/10.1016/0168-583X\(94\)95159-4](https://doi.org/10.1016/0168-583X(94)95159-4)
- [4] P. Ström, "Material characterization for magnetically confined fusion : Surface analysis and method development," Ph.D. dissertation, KTH, Fusion Plasma Physics, Stockholm, Sweden, Feb. 2019, qC 20190110.
- [5] M. Mayer, "Improved physics in simnra 7," *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, vol. 332, pp. 176–180, Aug. 2014, 21st International Conference on Ion Beam Analysis. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168583X14003139>
- [6] C. Jeynes and J. L. Colaux, "Thin film depth profiling by ion beam analysis," *Analyst*, vol. 141, no. 21, pp. 5944–5985, May 2016. [Online]. Available: <https://doi.org/10.1039/C6AN01167E>
- [7] M. Mayer, "Simnra, a simulation program for the analysis of nra, rbs and erda," *AIP Conference Proceedings*, vol. 475, no. 1, pp. 541–544, Apr. 1999. [Online]. Available: <https://aip.scitation.org/doi/abs/10.1063/1.59188>
- [8] J. F. Ziegler, M. Ziegler, and J. Biersack, "Srim – the stopping and range of ions in matter," *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, vol. 268, no. 11, pp. 1818–1823, Feb. 2010, 19th International Conference on Ion Beam Analysis. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168583X10001862>
- [9] —, *SRIM - The Stopping and Range of Ions in Matter*. Chester, MD: SRIM Co., 2008.
- [10] K. Arstila, J. Julin, M. Laitinen, J. Aalto, T. Konu, S. Kärkkäinen, S. Rahkonen, M. Raunio, J. Itkonen, J.-P. Santanen, T. Tuovinen, and T. Sajavaara, "Potku – new analysis software for heavy ion elastic recoil detection analysis," *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, vol. 331, pp. 34–41, Jul. 2014, 11th European Conference on Accelerators in Applied Research and Technology. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168583X14002717>
- [11] T. Konu, *User Manual for Potku*, University of Jyväskylä, Jyväskylä, Finland, Jul. 2013.
- [12] K. Arstila, T. Sajavaara, and J. Keinonen, "Monte carlo simulation of multiple and plural scattering in elastic recoil detection," *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, vol. 174, no. 1, pp. 163–172, Mar. 2001. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168583X00004353>
- [13] M. Mayer, *SIMNRA User's Guide*, Max-Planck-Institut für Plasma-physik, Garching, Germany, 2020.
- [14] S. Giangrandi, K. Arstila, B. Brijs, T. Sajavaara, A. Vantomme, and W. Vandervorst, "Considerations about multiple and plural scattering in heavy-ion low-energy erda," *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, vol. 267, no. 11, pp. 1936–1941, May 2009. [Online]. Available: <https://doi.org/10.1016/j.nimb.2009.03.105>

Comparison of RF Heating in ASDEX Upgrade and ITER

Axel Blennå and Mark Kalldas

Abstract—The increased effects of global warming have been a driving force to further research and develop sustainable energy sources, such as fusion. In this study, two different fusion devices are compared in terms of ion cyclotron resonance heating (ICRH) of plasma. The two devices are the tokamaks ASDEX Upgrade and not yet built ITER. To make the comparison, ICRH was simulated in the two tokamaks using the FEMIC code. ASDEX Upgrade was simulated with a deuterium plasma and ITER was simulated both with a deuterium and a deuterium-tritium plasma. In all scenarios a 3% minority species concentration, consisting of helium-3, was introduced. The obtained results show a higher and more centered wave absorption in ITER, compared to ASDEX Upgrade. This is mainly due to the size difference of the tokamaks. The smaller plasma radius of ASDEX Upgrade allowed for more wave reflection in the plasma, resulting in standing waves that formed eigenmode patterns. For simulations in ITER, the waves were absorbed before they could be reflected in the plasma. Instead of standing waves and eigenmode patterns, the waves behaved as beams, propagating in a narrow region of the plasma. This indicates that ITER is more effective in terms of ICRH, as the absorption is greater and more focused to the center, minimizing power losses to the surroundings.

Sammanfattning—De ökade konsekvenserna av den globala uppvärmningen har varit en drivkraft för fortsatt forskning och utveckling av hållbara energikällor, bland annat fusion. I den här studien jämförs två olika fusionsanläggningar med avseende på joncyklotronresonansuppvärmning (ICRH) av plasma. De två anläggningarna är tokamakerna ASDEX Upgrade och ännu inte byggda ITER. För att göra jämförelser simulerades ICRH i de två tokamakerna med hjälp av FEMIC-koden. ASDEX Upgrade simulerades med ett deuteriumplasma och ITER simulerades med både ett deuterium- och ett deuterium-tritiumplasma. För alla scenarier introducerades en 3% minoritetskoncentration av helium-3. Resultaten visar en högre och mer centrerad vågabsorption i ITER jämfört med ASDEX Upgrade. Detta beror framför allt på storleksskillnaden mellan tokamakerna. Den kortare plasmaradien av ASDEX Upgrade tillät mer reflektion i plasmat, vilket resulterade i stående vågor som bildade egenmodsmönster. För simuleringar i ITER absorberades vågorna innan de kunde reflekteras i plasmat. I stället för stående vågor och egenmodsmönster uppförde vågorna sig som strålar som propagerade över en smal region i plasmat. Det här indikerar att ITER är mer effektiv med avseende på ICRH, då absorptionen är större och mer centrerad, vilket minimerar effektförluster till omgivningen.

Index Terms—Fusion, ASDEX Upgrade, ITER, FEMIC, ICRH, eigenmode, beam, wave absorption

Supervisors: Thomas Jonsson and Björn Zaar

TRITA number: TRITA-EECS-EX-2022:153

I. INTRODUCTION

In this section, fusion energy is introduced and motivated as a relevant field of study. An explanation of the fusion reaction

is given as well as a description of a method of plasma heating. The tokamak device is then introduced along with examples of two such facilities relevant to the project. Finally, the goals of the project are stated.

A. Sources of Energy

The demand for energy is steadily rising as population and the standard of living increases worldwide. A majority of the energy consumed is generated by the use of fossil fuels, with coal and oil being the most common sources [1]. These non-renewable, fossil based energy sources lead to extensive emissions of carbon dioxide into the atmosphere, which increases the green house effect and is the main factor of the rising global temperature [2]. The increasing temperature has severe consequences for the planet's environment and inhabitants, such as more frequent wildfires and droughts, rising sea levels, and the potential extinction of certain species [3].

To reduce the emissions of green house gases and mitigate the effects of climate change, non-sustainable energy sources have to be replaced by sustainable ones. One such alternative is fusion energy, which is being intensely researched as a promising source of future energy. Fusion power relies on nuclear fusion, where two light atomic nuclei fuse together to form a heavier nucleus while energy is released in the process. This is in contrast to the fission based nuclear power of today, where heavy atoms are split into lighter ones to release energy [4].

The fusion reaction does not generate any green house gases, making it a clearly advantageous energy source over fossil fuels. The fuel that is planned to be used in the first fusion power plants is deuterium and tritium, which are both isotopes of hydrogen. Deuterium can be found in sea water while tritium, which is radioactive, can be produced from lithium, meaning fusion fuel can be made available as it is needed [4]. Additionally, the fusion process results in considerably less radioactive waste, and with a shorter half life, than nuclear fission [4], again giving fusion power the edge.

B. The Fusion Process

To create a fusion reaction, the nuclei of the reactants have to be sufficiently close together in order for the strong nuclear force to dominate over the repelling Coulomb force between the nuclei [5]. This can be achieved by heating the atoms to high temperatures, which increases their kinetic energy and makes it possible for the nuclei to get within the required distance from each other [6]. Because of the high temperatures,

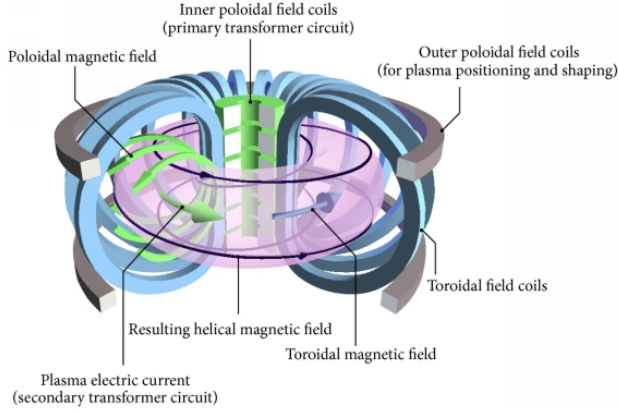


Fig. 1. A visualization of a tokamak's interior and magnetic field lines. The image is taken from [13].

the electrons are separated from their nuclei, ionizing the fusion fuel and creating a plasma where the fusion reactions can then occur [4].

C. The Tokamak

Due to its high temperature, the plasma cannot be allowed to touch any surfaces. One way to prevent this is by using a tokamak [7], a device shaped like a torus as seen in Fig. 1. The plasma is confined inside the tokamak using magnetic fields in the toroidal and poloidal directions. The toroidal field is induced by toroidal field coils [7] and poloidal magnetic fields are introduced by inducing a plasma current in the toroidal direction [8]. Together, the magnetic fields result in the helical field seen in Fig. 1. On the inner wall of the tokamak is also an antenna used for heating purposes [9]. Electromagnetic waves are transmitted by the antenna and propagate through the plasma where they are absorbed, heating the plasma. Under certain conditions, the waves can reach a natural state with standing waves that form an eigenmode pattern in the plasma. In other cases, the waves propagate more like a beam with a narrow shape.

There are many facilities around the world conducting fusion experiments. Among them are AUG (ASDEX Upgrade) and soon to be built ITER (International Thermonuclear Experimental Reactor), which are both of the tokamak type. ITER is planned to have a major radius of 6.2 metres [10], compared to AUG with a major radius of 1.6 metres [11], and will be the biggest tokamak in the world [10]. While AUG has a deuterium plasma [11], ITER will have a plasma consisting of both deuterium and tritium [12].

D. Plasma Heating

An important part of fusion research focuses on how to heat the plasma to reach and maintain sufficient temperatures to maximize the number of fusion reactions that occur. There are several possible methods, one of which is ion cyclotron resonance heating (ICRH). This method makes use of an antenna to transmit electromagnetic waves into the plasma

[9]. In the plasma, the ionized particles gyrate around the helical magnetic field lines with certain cyclotron frequencies. The helical magnetic field lines can be seen in Fig. 1. If the cyclotron frequency of a type of particle is the same as the frequency of the incoming waves from the antenna, the waves can be absorbed in the plasma and their energy transferred to the particles, thus heating the plasma [9]. For ICRH, the waves are made to resonate with the ions of the plasma, but resonances with the electrons of the plasma are also possible (ECRH).

E. Project Goals

The purpose of building fusion facilities such as AUG and ITER is research, but not all things need to be learned from experiments. Through simulations one can observe certain behaviors of waves and plasma that can be used as a good basis for real life experiments. In this project, these types of simulations will be made using the FEMIC code [14] and COMSOL® Multiphysics [15], which are tools that can be used to model ICRH in a plasma. The aim of this project is to examine what effects the larger size of ITER will have on plasma heating through ICRH. Therefore simulations in ITER will be performed and compared to the smaller tokamak AUG in order to examine the following:

- The differences between radio wave heating using ICRH in ASDEX Upgrade and ITER.
- Whether the transmitted waves from the antenna can be seen as an eigenmode or a beam.
- How and to what extent the eigenmode structure affects the heating of the plasma.
- How the impedance of the plasma differs between ASDEX Upgrade and ITER seen from the radio wave antenna's perspective.
- The difference in the distribution of waves in the toroidal direction between ASDEX Upgrade and ITER.

II. THEORY

The theoretical basis for the project is explained in this section. The propagation of electromagnetic waves in plasma is described first, followed by the dielectric tensor as well as the dispersion relation. The behavior of charged particles subjected to the magnetic field in the plasma is later explained, as well as the requirements for ion cyclotron absorption with regards to polarization. Finally, there is a description of plasma coupling.

A. Electromagnetic Waves in Plasma

Electromagnetic phenomena can be described by Maxwell's equations. In vacuum, these equations can be stated as follows [16]

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}, \quad (1)$$

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{J} + \frac{1}{c^2} \frac{\partial \mathbf{E}}{\partial t}, \quad (2)$$

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0}, \quad (3)$$

$$\nabla \cdot \mathbf{B} = 0, \quad (4)$$

where \mathbf{E} and \mathbf{B} are the electric and magnetic field vectors and \mathbf{J} is the current density. The permeability and permittivity in vacuum are denoted by μ_0 and ϵ_0 , respectively. Using the curl operator on Eq. (1) makes it possible to combine it with Eq. (2) into the wave equation,

$$\nabla \times (\nabla \times \mathbf{E}) + \frac{1}{c^2} \frac{\partial^2 \mathbf{E}}{\partial t^2} = -\mu_0 \frac{\partial \mathbf{J}}{\partial t}. \quad (5)$$

Solutions of Eq. (5) represent electromagnetic waves and can be analysed with the following form

$$\mathbf{E}(\mathbf{r}, t) = \Re[\mathbf{E}_0 e^{i(\mathbf{k} \cdot \mathbf{r} - \omega t)}], \quad (6)$$

which is a plane wave with angular frequency ω and wave vector \mathbf{k} . This assumption transforms the derivatives into

$$\frac{\partial}{\partial t} \rightarrow -i\omega \quad (7)$$

and

$$\nabla \rightarrow i\mathbf{k}. \quad (8)$$

Using the formulas for the current density [9]

$$\mathbf{J} = \boldsymbol{\sigma} \cdot \mathbf{E} \quad (9)$$

and the index of refraction vector [9]

$$\mathbf{n} = \frac{c}{\omega} \mathbf{k}, \quad (10)$$

the wave equation can be written as

$$\mathbf{n} \times (\mathbf{n} \times \mathbf{E}) + \mathbf{K} \cdot \mathbf{E} = 0. \quad (11)$$

In Eq. (11), \mathbf{K} is the dielectric tensor, which can be represented by a matrix made up of K_{ij} components. The dielectric tensor describes the plasma's effect on the propagating wave and is given by

$$\mathbf{K} = \mathbf{I} + \frac{i}{\epsilon_0 \omega} \boldsymbol{\sigma}, \quad (12)$$

where $\boldsymbol{\sigma}$ is the conductivity tensor [9]. The relationship between ω and \mathbf{k} is referred to as the dispersion relation and can be obtained by setting the determinant of Eq. (11) to zero. With a coordinate system according to [17], the following is obtained

$$\begin{vmatrix} K_{xx} - n_{\perp}^2 & K_{xy} & K_{xz} + n_{\perp} n_{\parallel} \\ -K_{xy} & K_{yy} - n_{\perp}^2 - n_{\parallel}^2 & K_{yz} \\ K_{xz} + n_{\perp} n_{\parallel} & -K_{yz} & K_{zz} - n_{\perp}^2 \end{vmatrix} = 0. \quad (13)$$

In Eq. (13) n_{\perp} and n_{\parallel} are the components of the index of refraction perpendicular and parallel to the toroidal axis and also to the magnetic field lines. An approximate solution for n_{\perp}^2 is

$$n_{\perp}^2 = K_{yy} - n_{\parallel}^2 + \frac{K_{xy}^2}{K_{xx} - n_{\parallel}^2}. \quad (14)$$

Using Eq. (10), the dispersion relation can then be obtained as

$$k_{\perp}^2 = \frac{\omega^2}{c^2} \left(K_{yy} - n_{\parallel}^2 + \frac{K_{xy}^2}{K_{xx} - n_{\parallel}^2} \right). \quad (15)$$

The dispersion relation is an approximation of the component of the wave number that is perpendicular to the magnetic field lines. In Eq. (15), there is a singularity when $K_{xx} = n_{\parallel}^2$. The singularity manifests as a thin layer in the plasma and is referred to as the ion-ion hybrid layer. The waves are expected to have a short wavelength near this layer in the plasma, as k_{\perp} becomes very large.

B. Ion Cyclotron Resonance Heating

Charged particles are subject to forces from electromagnetic fields. The contribution of the magnetic field \mathbf{B} to the force on an ion with charge q and velocity \mathbf{v} is according to Lorentz force equation [18]

$$\mathbf{F} = q\mathbf{v} \times \mathbf{B}. \quad (16)$$

The force in Eq. (16) makes the ions gyrate around the helical magnetic field lines in the plasma. The cyclotron frequency with which the ions gyrate is given by

$$\omega_c = \frac{|q||\mathbf{B}|}{m}, \quad (17)$$

where m is the mass of the ion. If the poloidal field in Fig. 1 is ignored, the magnetic field in the plasma can be approximated as

$$|\mathbf{B}| = \frac{B_0 R_0}{R}, \quad (18)$$

where R is the radial coordinate of the tokamak and the index 0 denotes the center of the plasma [8]. Using Eq. (18) in Eq. (17) makes it possible to calculate the cyclotron frequency at a specific distance from the tokamak center. For heating purposes, it is desirable for the waves to resonate with the ions at the center of the plasma. This can be achieved by setting the angular frequency ω of the transmitted wave from the antenna equal to the ion cyclotron frequency ω_c in the middle of the plasma. When $\omega = \omega_c$, resonance occurs and the ions in the middle of the plasma can absorb the energy from the transmitted waves. However, this does not happen at the fundamental frequency for a plasma consisting of only one ion species [9]. One way to achieve absorption at the fundamental frequency is by introducing a minority species according to Section II-D.

C. Polarization

For a plane wave, the direction of the electric field can be expressed by polarization. A wave can be linearly, elliptically or circularly polarized. For a linearly polarized wave, the electric field only oscillates in one direction. The combination of two linearly polarized waves form an elliptically polarized wave. If the linearly polarized waves are orthogonal to each other, have equal amplitude and are phase shifted by 90° relative to each other, the polarization will be circular.

If the electric field rotates in a counterclockwise direction relative to the direction of the magnetic field, the wave is right-hand polarized. This is also referred to as negative circular polarization. If the electric field instead has a clockwise rotation, and the direction of the magnetic field remains the same, the wave is left-hand polarized. This is known as positive circular polarization [19]. Note that for this project it is more relevant to look at the propagating wave from the perspective of a receiver and for the electric field to be described relative to the magnetic field. This is different to [19], where the electric field is described relative to the direction of wave propagation, viewed from a source. In this project, the left-hand and right-hand polarized electric fields can therefore be expressed by

$$E_+ = \frac{E_x + iE_y}{2} \quad (19)$$

and

$$E_- = \frac{E_x - iE_y}{2}, \quad (20)$$

when \mathbf{B} is in the z -direction.

D. Introducing a Minority Species

A transmitted wave from an antenna can in general be described as an elliptically polarized wave, formed as a sum of two waves with positive and negative circular polarization. In the case of ion cyclotron absorption in a plasma, only positive circular polarization is desired [20]. Negative circularly polarized waves have an electric field that rotates in the opposite direction to the gyro motion of the positive ions, moving according to Eq. (16), thus resulting in no ion cyclotron absorption. For a positive circularly polarized wave, the direction of the electric field is instead parallel to the gyro motion of the ions, which results in ion cyclotron absorption.

The ratio between left-hand and right-hand polarized electric fields in a deuterium plasma can approximately be written as [9]

$$\left| \frac{E_+}{E_-} \right| \approx \left| \frac{\omega - \omega_c}{\omega + \omega_c} \right|, \quad (21)$$

where ω is the angular frequency of the transmitted wave from the antenna and ω_c is the ion cyclotron frequency. According to Eq. (17), the cyclotron frequency for an ion is dependant on its radial coordinate, R . Ions in different parts of the plasma therefore have different cyclotron frequencies. To ensure that resonance occurs in the center of the plasma, the resonance frequency is set equal to the cyclotron frequency for ions in the center. However, when resonance occurs at the fundamental frequency for a plasma consisting of a single ion species, E_+ in Eq. (21) is zero, which results in no ion cyclotron absorption. One way to achieve absorption at the fundamental frequency is by introducing a minority of a different ion species. The antenna frequency can be set to match the cyclotron frequency for the minority species in the center of the plasma [9]. One such example can be found in Fig. 2, where the chosen minority species is helium-3. E_+ in Eq. (21) is no longer zero at the resonance when the minority species is introduced, which results in absorption of the wave energy and leads to the heating of the minority species. The minority species then transfers energy to the rest of the plasma, heating it up.

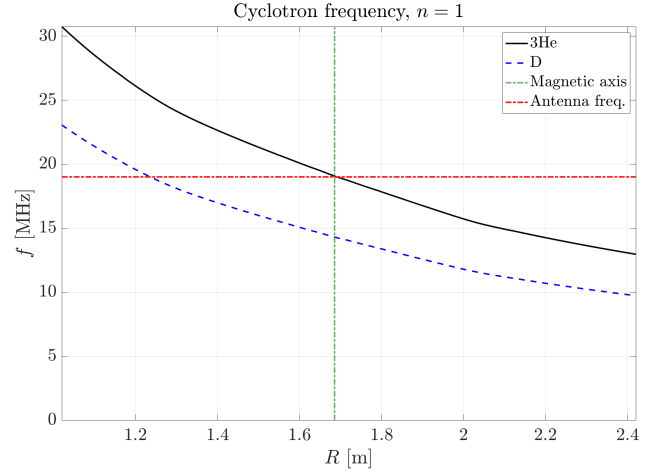


Fig. 2. The cyclotron frequencies in a deuterium plasma, where the minority species helium-3 has been introduced. The antenna frequency has been set to match the cyclotron frequency of helium-3 in the center of the plasma. $n = 1$ refers to the fundamental frequency.

E. Fourier Expansion of the Electric Field

Because of symmetry in the toroidal direction of the tokamaks, the total electric field can be represented as a sum of Fourier components. This is used to numerically calculate the electric field in the geometry of the tokamak. The Fourier sum can be written as

$$\mathbf{E}(R, \phi, Z) = \sum_{n_\phi=-\infty}^{\infty} \hat{\mathbf{E}}_{n_\phi}(R, Z) e^{in_\phi \phi}, \quad (22)$$

where n_ϕ is the toroidal mode number and should not be confused with the index of refraction. The electric field $\hat{\mathbf{E}}_{n_\phi}(R, Z)$ is a solution to Eq. (5) in the RZ -plane for a specific value of n_ϕ . The toroidal mode number represents the number of oscillations per cycle around the tokamak for each component. Individual simulations can be made for each of these components of the electric field.

F. Plasma Coupling

The electromagnetic waves propagate through the plasma and get absorbed by the minority species. The relationship between total absorbed power in the plasma, referred to as coupled power, and the antenna's current density can be described by:

$$P = |J|^2 R_P, \quad (23)$$

where R_P is the plasma impedance seen from the antenna's perspective. Note that the unit of R_P in Eq. (23) is $\Omega \text{ m}^2$.

Between the plasma and the tokamak's antenna is a vacuum region called the scrape off layer (SOL). In this region, the radio waves from the antenna are evanescent, meaning that they are exponentially decaying. This can be seen by considering the square of the amplitude of the wave vector, which in vacuum is given by

$$k^2 = k_\perp^2 + k_\parallel^2 = \frac{\omega^2}{c^2}. \quad (24)$$

The wave length and decay length of the wave can be described by $\Re\{k\}$ and $\Im\{k\}$, respectively. Solving Eq. (24) for k_{\perp}^2 gives

$$k_{\perp}^2 = \frac{\omega^2}{c^2} - k_{\parallel}^2. \quad (25)$$

For the most important wave modes, $k_{\parallel}^2 \gg \omega^2/c^2$ and the following approximation can be made

$$k_{\perp} \approx \pm i k_{\parallel}. \quad (26)$$

Since the component of the wave vector perpendicular to the magnetic field lines is imaginary in the SOL, the waves propagating from the antenna toward the plasma are exponentially decaying with a decay length $1/\Im(k_{\perp})$ seeing as

$$E \sim e^{-\Im\{k_{\perp}\}x}, \quad (27)$$

where x is the distance of propagation. To achieve strong coupling of the waves to the plasma, the antenna therefore has to be placed close to the plasma edge.

The wavelength of the parallel wave component is given by

$$\lambda = \frac{2\pi R}{n_{\phi}}, \quad (28)$$

where $2\pi R$ is the toroidal circumference for a given value of the radius R and n_{ϕ} is the toroidal mode number, which denotes the number of oscillations per cycle around said circumference. The parallel wave vector component can then be given by

$$k_{\parallel} = \frac{2\pi}{\lambda} = \frac{n_{\phi}}{R}. \quad (29)$$

A higher toroidal mode number n_{ϕ} results in a larger parallel wave vector component k_{\parallel} , hence a higher imaginary perpendicular wave component k_{\perp} according to Eq. (26). This will lead to a weaker electric field in the plasma due to the shorter decay length of the waves in the SOL. Because of this, the power coupled to the plasma will be smaller.

G. Reflection

The size of a tokamak will directly affect the amount of wave reflection that occurs in the plasma. The wave reflection will in turn affect the resulting wave patterns. Strong wave reflection and weak wave damping result in standing waves that form eigenmode patterns. If there is no wave reflection, there will be no formation of standing waves and thus the waves will behave as beams. One way to evaluate the amount of wave reflection is by considering the reflection coefficient [21]

$$|\Gamma| = \frac{S - 1}{S + 1}, \quad (30)$$

where S is the standing wave ratio (SWR) according to

$$S = \frac{|E_{\max}|}{|E_{\min}|}. \quad (31)$$

III. METHOD

First, a short overview of the wave solver FEMIC is given. There is also an explanation of how an appropriate minority concentration was determined. Thereafter, the simulations that have been performed are described, where the toroidal mode number was varied and 3D visualizations of the electric fields were created. Finally, it is described how the impedance of the plasma and the reflection coefficient were calculated.

A. The FEMIC Code

In order to compare the two tokamaks AUG and ITER in terms of ICRH, a number of simulations were made using the FEMIC code. FEMIC is a code based on MATLAB[®] [22] that simulates ICRH in a fusion plasma by using the finite element method. The code uses the software COMSOL[®] Multiphysics to solve the wave equation and post-processing the results of the simulations. The values of the parameters used for all simulations can be found in Table I, where B_0 and R_0 denote the magnetic field and the radius at the plasma center, respectively, as described in Eq. (18). The temperature and density profiles for both tokamaks can be found in Fig. 3a and 3b. It should be noted that the magnetic field in the poloidal direction was neglected in all simulations.

B. Choosing a Minority Species Concentration

Both AUG and ITER were simulated with a deuterium plasma, which also contained a minority species. The minority species chosen for this project was helium-3, as it will be the main minority species used in ITER [23].

For the modeling of ICRH in AUG and ITER, the antenna frequency had to be determined. To do this, the magnetic field was plotted to determine its value at the plasma center in each tokamak. This value was then used in Eq. (17) to determine the cyclotron frequency of helium-3 in the plasma center for the two tokamaks. The frequency of the antenna was then set to $f = \omega_c/2\pi$ for the simulations in FEMIC. The antenna frequencies used were 19 MHz in AUG and 53 MHz in ITER.

Simulations were made of the electric field in both facilities for helium-3 concentrations between 0% and 5%. The toroidal mode number was kept constant. For higher concentrations of the minority, the ion-ion hybrid layer can create short wavelengths that cannot be resolved by FEMIC. The simulations were therefore examined to determine at which concentration FEMIC was able to resolve the short wavelengths created by the ion-ion hybrid layer. The simulations showed that the waves near the hybrid layer could be fully resolved in the two tokamaks when a 3% minority concentration was added. This concentration was therefore chosen for all remaining simulations and will most likely be the concentration used in ITER [23].

TABLE I
PARAMETER VALUES USED IN SIMULATIONS FOR BOTH TOKAMAKS.

	Antenna Frequency [MHz]	B_0 [T]	R_0 [m]
AUG	19	1.8619	1.6855
ITER	53	5.3	6.2

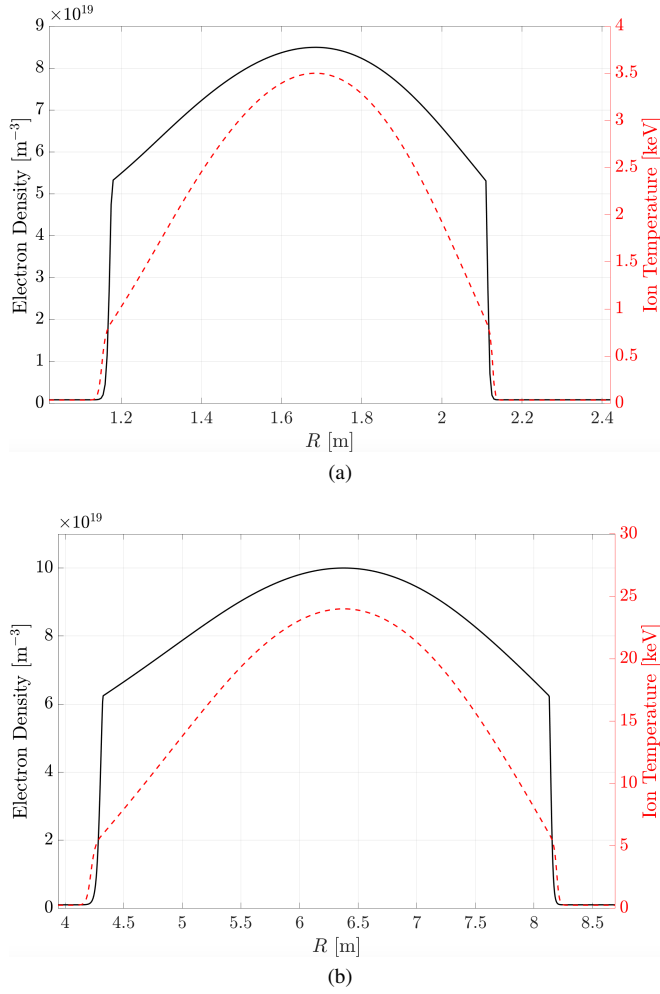


Fig. 3. Ion temperature (dashed line) and electron density (solid line) of the simulated plasmas in (a) AUG and (b) ITER.

C. Varying the Toroidal Mode Number

After the minority concentration for the plasma had been decided, simulations of the wave propagation for varying toroidal mode numbers n_ϕ were performed. Electric field components corresponding to different values of n_ϕ in Eq. (22) were simulated separately. The simulations for ITER were made both for a deuterium plasma as well as for a deuterium-tritium plasma. In AUG, only a deuterium plasma was considered.

To decide which toroidal mode numbers to simulate, the current density of the antenna in each tokamak was examined. A plot of the current spectrum as a function of n_ϕ for the antennas in AUG and ITER can be seen in Fig. 4. As the power delivered to the plasma depends on the antenna current according to Eq. (23), the values of n_ϕ where the current density had its largest peaks were the most relevant to simulate. In AUG, simulations were made for values of n_ϕ between 6 and 50. ITER, which is a bigger facility, was simulated with more toroidal modes, ranging from 25 to 71. Due to a lack of resolution near the ion-ion hybrid layer, lower toroidal mode numbers were not included in the simulations.

The simulations of ITER required considerably more time to complete than the simulations of AUG. To save time, the

deuterium plasma in ITER was only simulated for every other n_ϕ for the values between the peaks in the current spectrum. For the simulations of AUG and the deuterium-tritium plasma in ITER, every value of n_ϕ in the chosen intervals were included.

D. Creating 3D Fields

One of the goals of the project was to compare the distribution of waves in the toroidal direction of AUG and ITER. To accomplish this, the electric field for every simulated toroidal mode was scaled by the antenna current density for its corresponding mode in Fig. 4. The scaled electric fields for all modes were summed using Eq. (22) to create 3D fields representing the total electric field in the tokamak. As discussed in Section II-D, only positive circular polarization results in ion cyclotron absorption. It was therefore more relevant to look at the electric field component $E_+(R, \phi, Z)$, when comparing ICRH in AUG and ITER.

Electric field components for values of n_ϕ that had not been simulated were taken from the field component with the closest available value. The negative values in the sum were not simulated, but were taken from the corresponding positive values. For AUG, the sum was calculated for values of n_ϕ between -50 and 50 . The missing field components for the values of n_ϕ between 0 and 5 were taken from the field component for mode 6 . The sum for ITER was calculated for values of n_ϕ between -147 and 147 . Missing values were taken from available components in the same way as for AUG.

E. Calculating the Plasma Impedance

For all simulations, a current density on the antenna conductor of $J = 1 \cdot e^{in_\phi\phi} \text{ A m}^{-1}$ was used. Because of this, Eq. (23) relating the plasma impedance to the absorbed power becomes

$$P = [1 \text{ A m}^{-1}]^2 \cdot R_P. \quad (32)$$

This means that the impedance R_P can be determined from the amount of absorbed power in the plasma for each simulation.

F. Calculating the Reflection Coefficients

As a measure of the level of beam or eigenmode pattern in the two tokamaks, the reflection coefficient was calculated. This was done by considering the standing wave ratio of the electric field component E_- and calculating the reflection coefficient according to Eq. (30). The calculations were made for a few different values of n_ϕ in both AUG and ITER.

IV. RESULTS

In this section, the wave propagation in both AUG and ITER are shown for selected values of the toroidal mode number n_ϕ . The results are examined in terms of the theoretical dispersion relation for each case. The polarization in ITER and a description of the ion absorption in both tokamaks is later given. To illustrate the electric field distribution in the toroidal direction in the two tokamaks, 3D visualizations of the electric fields are presented. Followed by that, the plasma impedance, as defined in Section II-E, is evaluated as a function of the toroidal mode number. Finally, the reflection coefficient for different toroidal mode numbers is given.

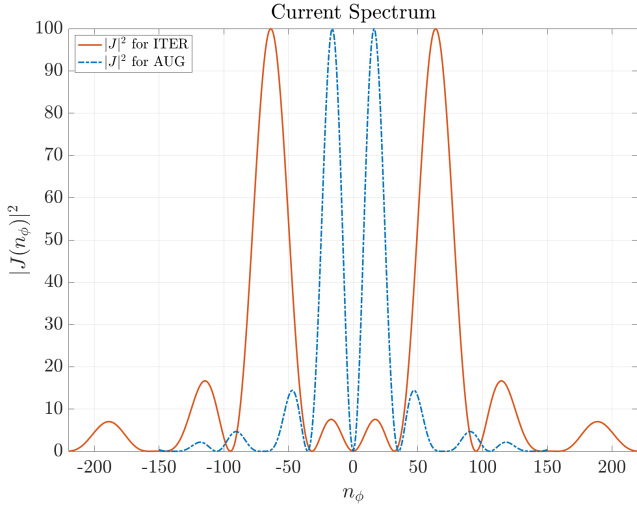


Fig. 4. Current density as a function of n_ϕ for the antennas in AUG and ITER, respectively.

A. Wave Propagation in AUG

To demonstrate the differences in wave propagation for varying toroidal mode numbers, the left-hand component E_+ of the electric field in AUG has been plotted for three modes. The corresponding plots for $n_\phi = 10, 20$ and 50 can be seen in Fig. 5a, 5b and 5c. The figures depict slices of the torus-shaped tokamak at $\phi = 0$. The ICRH antenna is located on the right hand side in each figure, meaning that the waves are mainly propagating from right to left. The electric field behaves as a standing wave and forms an eigenmode pattern in Fig. 5a and 5b. The magnitude of the field is higher for $n_\phi = 10$ than $n_\phi = 20$. In Fig. 5c where $n_\phi = 50$, the magnitude is small and the wave decays quickly and disappears before it reaches the plasma center.

The vertical line stretching across the middle of the plasma in Fig. 5a is the ion-ion hybrid layer. The color scale for the hybrid layer is fully saturated for $n_\phi = 10$, but barely noticeable in Fig. 5b for $n_\phi = 20$. It has completely disappeared for $n_\phi = 50$ in Fig. 5c.

To further understand the behavior of the waves, the dispersion relation in AUG for the different values of n_ϕ is displayed in Fig. 7a. The real part of the wave vector component k_\perp describes the inverse wave length perpendicular to the magnetic field lines, i.e. in the (R, Z) plane. As described in Section II-A, there is a singularity in the dispersion relation. The singularity causes the peaks at $R \approx 1.6$ m. The peaks disappear for higher n_ϕ , when the ion-ion hybrid layer disappears.

B. Wave Propagation in ITER

Similarly to AUG, the left-hand component of the electric field in ITER has been plotted for two different values of n_ϕ for a deuterium-tritium plasma. The electric field for $n_\phi = 25$ can be seen in Fig. 6a, while Fig. 6b shows the field for $n_\phi = 70$. For $n_\phi = 25$, the ion-ion hybrid layer is clearly visible as the fully saturated white area in the middle of the plasma and

the faint red lines extending in the Z-direction. This effect is almost completely gone for $n_\phi = 70$.

The dispersion relation is visualised in Fig. 7b, where the real part of the wave vector component k_\perp has been plotted for the two cases. The ion-ion hybrid layer introduces the peaks in the dispersion relation at the radius where the singularity is located. As can be seen by the peaks in Fig. 7b, the ion-ion hybrid layer is considerably more prominent for $n_\phi = 25$ compared to $n_\phi = 70$.

As seen in Fig. 6a and Fig. 6b, the waves in ITER, unlike in AUG, behave as a beam rather than forming a pattern of eigenmodes. For $n_\phi = 70$, the beam has a slightly narrower shape and the electric field strength is lower than for $n_\phi = 25$. A weaker electric field strength at the minority resonance means that less power is absorbed by the plasma for the higher toroidal mode number. This is further confirmed by looking at the dispersion relation for the two cases. The waves propagate from right to left, through the SOL indicated by the grey area in Fig. 7b. As the real part of k_\perp is nonzero in the SOL for $n_\phi = 25$, the waves are able to propagate from the antenna to the plasma in that case. For $n_\phi = 70$ however, the real part of k_\perp is zero, meaning the waves are exponentially decaying from the antenna to the plasma. Because of this, less wave energy is able to reach the plasma for the higher toroidal mode number.

C. Polarization in ITER

The square of the ratio between E_+ and the total electric field amplitude in ITER has been plotted in Fig. 8 for both a deuterium as well as a deuterium-tritium plasma. In the center of the plasma, there is a sudden peak in the polarization caused by the ion-ion hybrid layer. The green dashed line to the right in the figure indicates the minority resonance. The ratio at the minority resonance is slightly lower for the deuterium plasma compared to the deuterium-tritium plasma, indicating a lower magnitude of E_+ in that case. Since E_+ is the desired electric field component to achieve resonance, as explained in Section II-D, the results in Fig. 8 further suggest a somewhat lower power absorption in the deuterium plasma compared to the deuterium-tritium plasma.

D. Absorbed Power

The propagating waves transfer power to the minority species helium-3 at the resonance position. To get a better understanding of resonance heating, it is therefore of interest to examine Fig. 9a, 9b and 9c, which show the power absorption of helium-3 in AUG for $n_\phi = 10, 20$ and 50 , respectively. The power absorption decreases for increasing values of n_ϕ . This is in accordance with the results of the wave propagation in AUG, which showed a lower amplitude of the electric field for higher toroidal mode numbers. The power absorption for $n_\phi = 10$ and 20 is mostly focused to the center of the plasma, where the minority resonance occurs. For $n_\phi = 50$, Fig. 9c shows a very low power absorption that occurs before the minority resonance. This can be explained by the wave propagation in Fig. 5c and the dispersion relation in Fig. 7a,

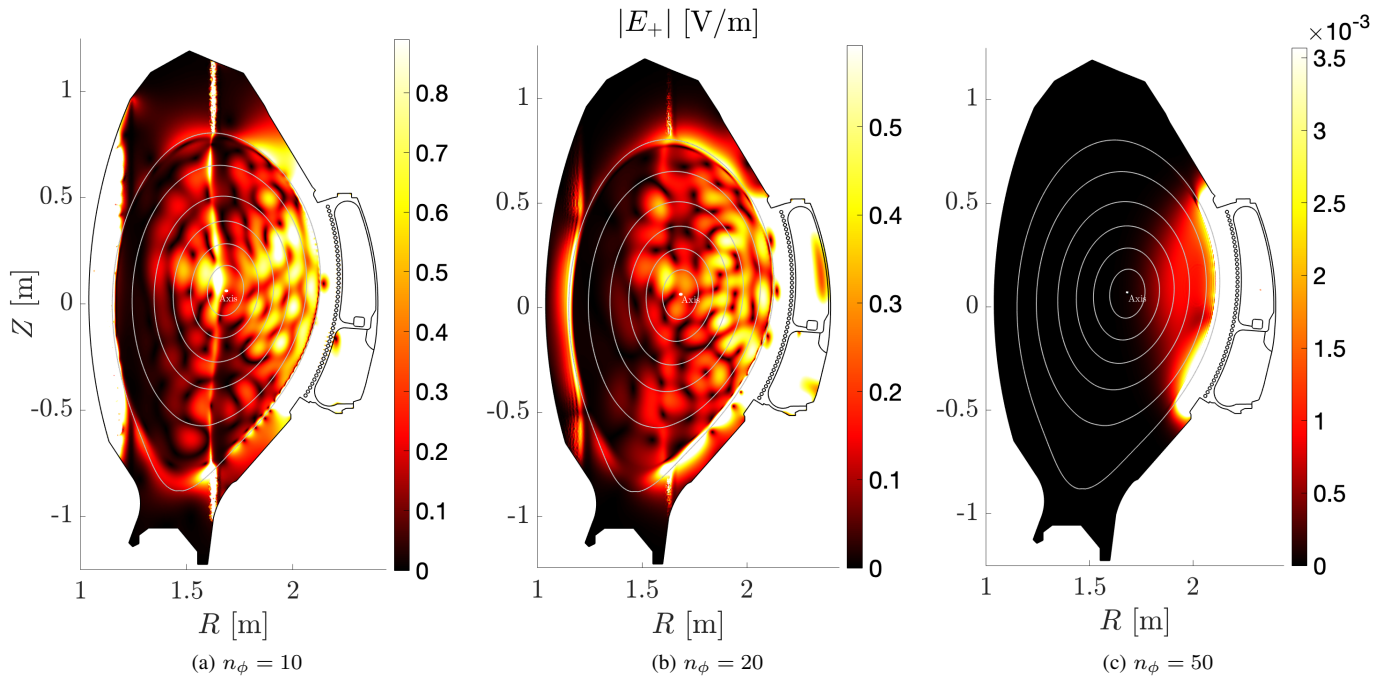


Fig. 5. Magnitude of the left hand component of the electric field, $|E_+|$ [V/m], for the toroidal mode numbers 10, 20 and 50 in a deuterium plasma in AUG.

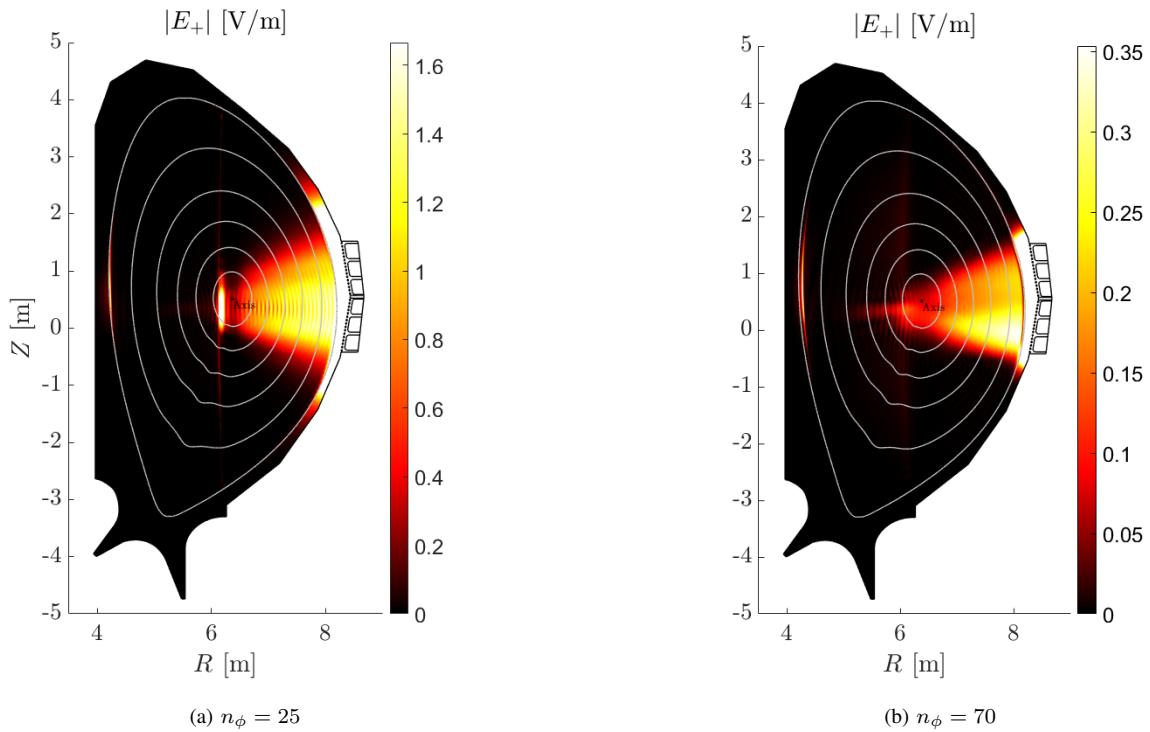


Fig. 6. Magnitude of the left hand component of the electric field, $|E_+|$ [V/m], for the toroidal mode numbers 25 and 70 in a deuterium-tritium plasma in ITER.

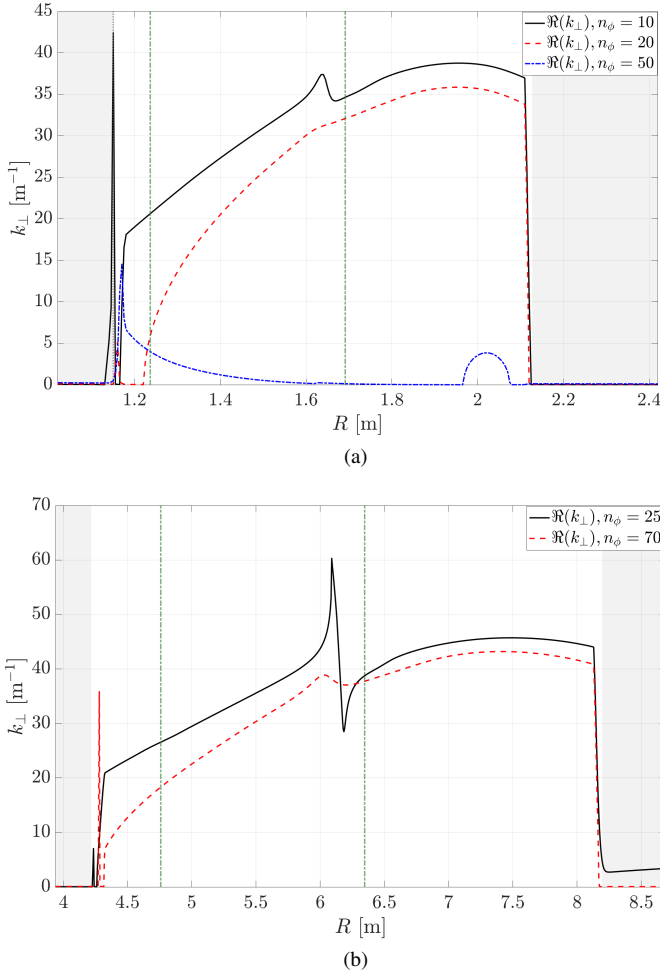


Fig. 7. Dispersion relation for different toroidal mode numbers in a deuterium plasma in AUG and a deuterium-tritium plasma in ITER. The green dashed lines indicate the radius where resonance occur. The right one is for the minority resonance and the left for the deuterium resonance.

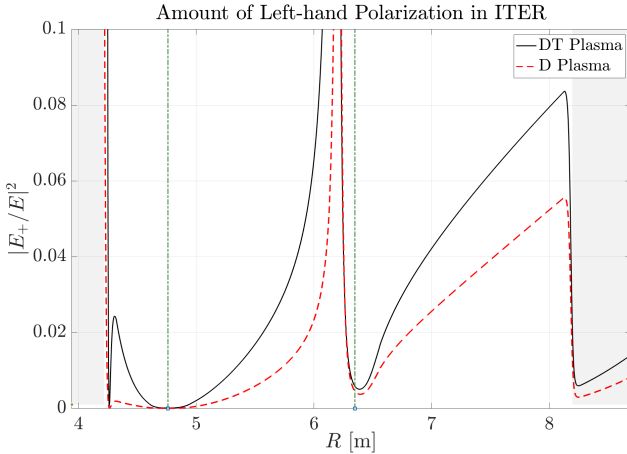


Fig. 8. Squared ratio of the left-hand polarized component and the total electric field amplitude in ITER for $n_{\phi} = 25$. The green dashed lines indicate the radius where resonance occur. The right one is for the minority resonance and the left for the deuterium resonance.

where it can be seen that the waves stop propagating before they reach the plasma center.

The power absorption of helium-3 in ITER is shown in

Fig. 10a for $n_{\phi} = 25$ and in Fig. 10b for $n_{\phi} = 70$. There is considerably more power absorbed for the lower toroidal mode number. The absorption is also shaped like a pattern of interference, compared to the higher toroidal mode number where the absorption is more spread out and the pattern not as apparent.

E. 3D Representation of the Electric Field

To get a more complete view of the wave propagation in the two tokamaks, the simulated electric field components for different values of n_{ϕ} were scaled by the antenna current and summed to yield the total electric field, according to Section III-D. The real part of $E_+(R, \phi, Z)$ was then plotted in planes of both tokamaks to create 3D visualizations of the wave propagation. Each plane is a slice of the tokamak, either for a constant value of z or ϕ , and depicts the wave propagation in said plane.

The distribution of waves in the toroidal direction for a deuterium plasma in AUG can be seen in Fig. 11a and 11b. In both figures, the waves form an eigenmode pattern that spreads throughout the entire plasma. The field is strongest near the antenna, before the waves reach the ion-ion hybrid layer. The hybrid layer stretches around the entire tokamak, but decays the further it gets from the antenna.

The toroidal wave distribution in ITER for a deuterium-tritium plasma can be seen in the 3D field in Fig. 12. The waves form a beam from the antenna into the plasma and are not spread out to other parts of the tokamak. The majority of the wave energy is absorbed in the middle of the plasma in front of the antenna.

F. Plasma Impedance

As discussed in Section III-E, the plasma impedance seen from the radio wave antenna's perspective is proportional to the coupled power, which results in resonance heating. The plasma impedance for different values of n_{ϕ} in AUG has therefore been depicted in Fig. 13a. The impedance exponentially decreases for higher values of n_{ϕ} . However, there is a spike in intervals of four or five in n_{ϕ} , where there is a significant increase in coupled power.

The plasma impedance in ITER has been plotted in Fig. 13b for deuterium and deuterium-tritium. Similarly to AUG, the impedance of both plasmas in ITER show an exponential decrease as the value of n_{ϕ} increases. As seen in the figure, the deuterium plasma in ITER has slight oscillations in impedance for varying toroidal mode numbers. This effect is not observed in the impedance graph of the deuterium-tritium plasma. Aside from the oscillations, the impedance values are similar in the two ITER plasmas.

G. Reflection Coefficients

To compare the amount of reflection in both tokamaks, the reflection coefficients were calculated for different values of n_{ϕ} in AUG and ITER, as seen in Tables II and III, respectively. The SWR, which was used to calculate the reflection coefficients, was approximately determined according to Eq. (31).

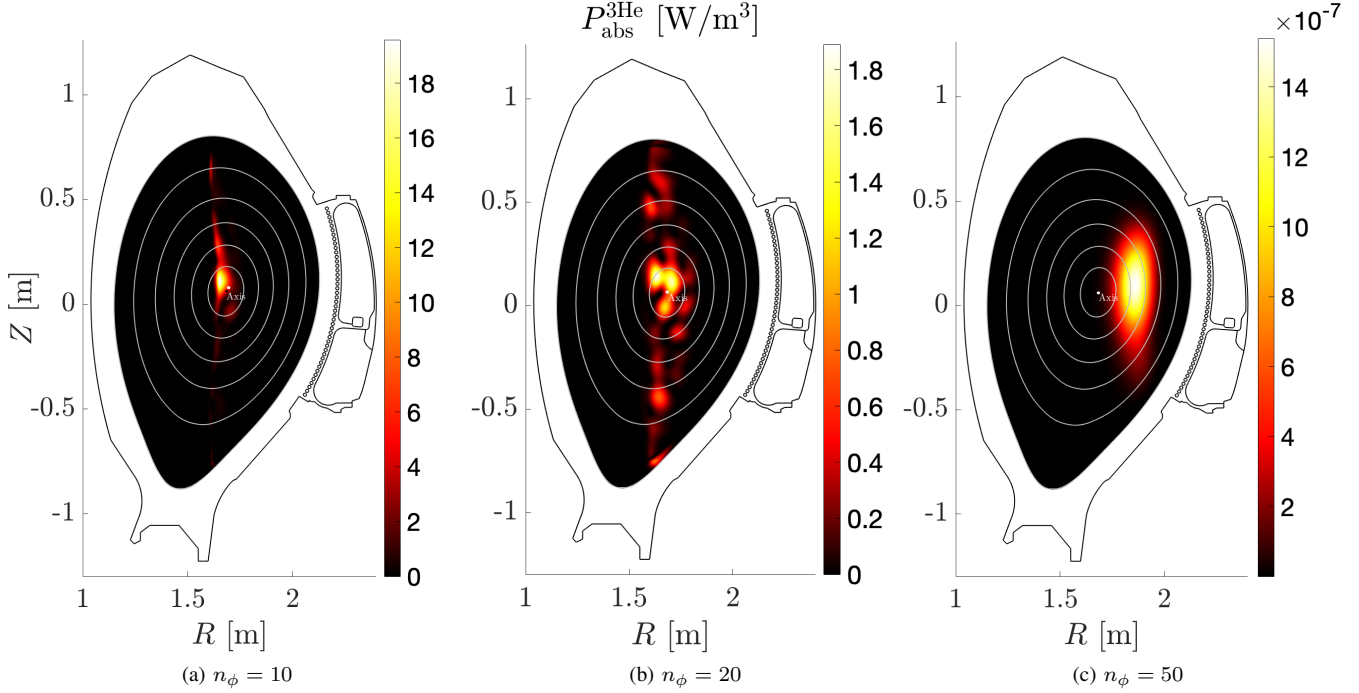


Fig. 9. Power absorption [W/m^3] of helium-3 for the toroidal mode numbers 10, 20 and 50 in a deuterium plasma in AUG.

The maximum and minimum of the electric field component E_- were observed to give estimated values of the SWR. The geometry of the two tokamaks allowed for wave reflection in different parts of the plasma, causing waves to interact with each other at different phase values. There were therefore large variations in E_- , which contained several maximum and minimum points with different amplitudes. This was more apparent in simulations of AUG, with a smaller major radius and more reflection at the plasma edges. Therefore, the calculated values of the SWR and the reflection coefficients in Tables II and III are uncertain and have for the most part only been presented to illustrate the overall difference in wave reflection between the tokamaks.

The wave reflection seems to be ten times higher in AUG compared to the reflection in ITER, when simulated with a deuterium plasma. There are also differences in wave reflection between the two plasmas in ITER. For $n_\phi = 25$, the reflection coefficient for the deuterium-tritium plasma is approximately double that of the deuterium plasma. For the toroidal mode numbers 40 and 70, the SWR is instead larger for the deuterium plasma with a Γ around a factor ten larger than for deuterium-tritium. The values in Table III show the largest SWR, and therefore the largest Γ , for the lowest toroidal mode number in both plasmas.

V. DISCUSSION

The obtained results and their implications are discussed in this section. First, the propagation of waves in the two tokamaks is compared in terms of wave pattern and the ion-ion hybrid layer. The power absorption by the minority is then discussed and linked to whether the waves behave as a beam or a pattern of eigenmodes. Next, the plasma impedance in

TABLE II
STANDING WAVE RATIO (SWR) AND REFLECTION COEFFICIENT $|\Gamma|$ IN AUG FOR DIFFERENT VALUES OF THE TOROIDAL MODE NUMBER n_ϕ .

n_ϕ	SWR	$ \Gamma $
10	3	0.5
15	2.5	0.43
20	3.5	0.56
25	2.5	0.43
30	6	0.71

TABLE III
STANDING WAVE RATIO (SWR) AND REFLECTION COEFFICIENT $|\Gamma|$ IN ITER FOR DIFFERENT VALUES OF THE TOROIDAL MODE NUMBER n_ϕ .

n_ϕ	SWR (DT)	$ \Gamma $ (DT)	SWR (D)	$ \Gamma $ (D)
25	1.27	0.12	1.16	0.074
40	1.01	0.0032	1.07	0.032
70	1.02	0.0087	1.16	0.072

AUG and ITER is compared, with a discussion on how it is affected by the wave propagation in the tokamaks. Finally, there is a discussion on the 3D fields and how they differ.

A. Wave Propagation

One of the goals of the project was to compare AUG and ITER in terms of heating through ICRH. It was therefore of interest to analyse and compare the propagation of electromagnetic waves in the two tokamaks. In AUG, the propagating waves formed eigenmode patterns, which were stronger for lower values of n_ϕ . The waves in ITER behaved as beams and propagated in a narrower region of the plasma. The differences in wave propagation for the two tokamaks were largely due to the difference in size. The smaller size of AUG allowed

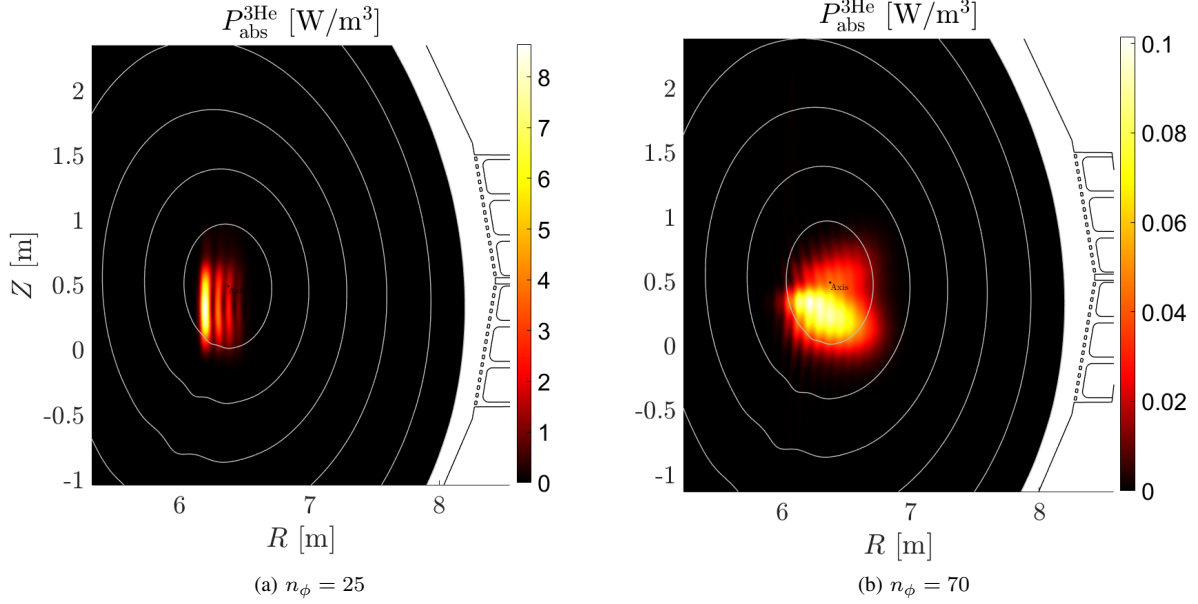


Fig. 10. Power absorption [W/m^3] of helium-3 for the toroidal mode numbers 25 and 70 in a deuterium-tritium plasma in ITER.

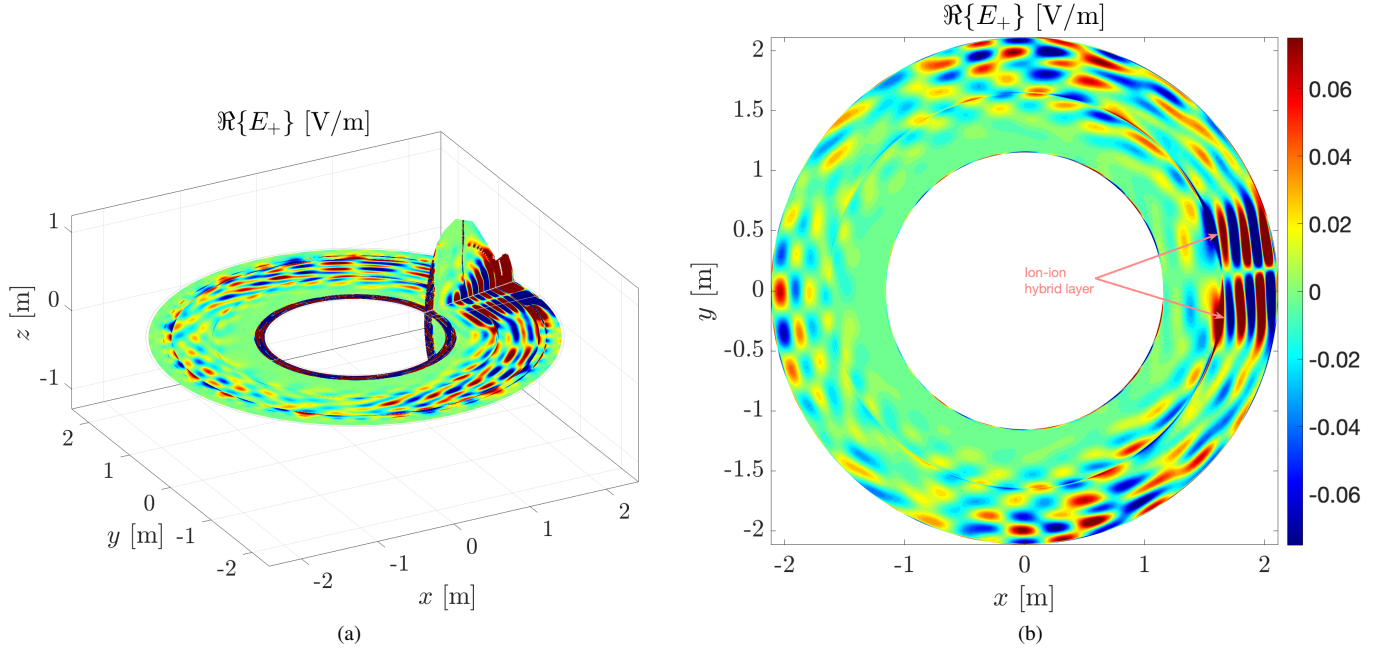


Fig. 11. Wave propagation in AUG for (a) two planes intersecting at $z = 0.13 \text{ m}$, $\phi = 0.2 \text{ rad}$ and for (b) one plane at $z = 0 \text{ m}$. Note that the saturation has been altered to include parts with lower magnitude.

for more wave reflection, as seen in Tables II and III, which resulted in standing waves that formed the eigenmode patterns in Fig. 5a and 5b. The larger size of ITER allowed the waves to propagate across a longer distance. A larger fraction of the wave energy in ITER could therefore be absorbed by the plasma before the waves could be reflected off the outer edges of the plasma, compared to AUG. This resulted in less wave reflection, thus no eigenmode pattern was formed. Instead, the waves behaved as beams, as seen in Fig. 6a and 6b.

Also, the plasma composition was observed to affect the

reflection of waves. For most toroidal mode numbers in ITER, there was less reflection in the deuterium-tritium plasma compared to the deuterium plasma, which can be seen in Table III. This is due to higher wave absorption, and therefore stronger damping of the waves, in a deuterium-tritium plasma.

In both AUG and ITER, the ion-ion hybrid layer was visible for lower values of n_ϕ . It should be noted that what is considered to be low values of n_ϕ differs between the two tokamaks. As explained in Section II-F, n_ϕ denotes the amount of oscillations per cycle around the tokamak for a

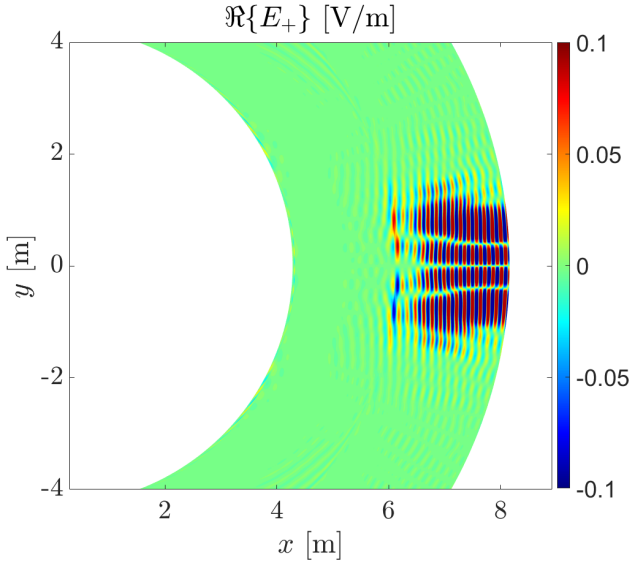


Fig. 12. Wave propagation in the horizontal plane, at $z = 0$ m, for a deuterium-tritium plasma in ITER. Note that the saturation has been altered to include parts with lower magnitude.

given radius. In order for ITER, with a larger circumference, to have the same parallel wavelength as AUG, n_ϕ must be chosen higher according to Eq. (28). This is why the ion-ion hybrid layer is very prominent for $n_\phi = 25$ in ITER, when it has almost completely disappeared in AUG for $n_\phi = 20$. The parallel wavelength at the plasma radius for both tokamaks is approximately the same when $n_\phi = 20$ in AUG and $n_\phi = 70$ in ITER.

The ion-ion hybrid layer, which is more prominent for a wider range of n_ϕ values in ITER, can cause waves to reflect off it. This could explain the higher rate of wave reflections for lower toroidal mode numbers n_ϕ in Tables II and III. The effects of the hybrid layer is more apparent in ITER, which can be seen by comparing Fig. 5a and 5b for AUG with Fig. 6a for ITER, but also by looking at the peaks in the dispersion relation in Fig. 7a and 7b. Fig. 7a also shows almost no propagation for $n_\phi = 50$ in AUG, which explains why the waves do not reach the plasma center in Fig. 5c. This also holds for larger values of n_ϕ in ITER.

B. Absorbed Power

Since the antenna in each tokamak was set to resonate with the minority of helium-3 ions in the center of the plasma, this minority was the main source of power absorption. Absorption at the center of the plasma is desirable for heating purposes. The power absorption of the helium-3 minority in AUG is mostly concentrated to the middle of the plasma for the toroidal mode numbers 10 and 20, as shown in Fig. 9a and 9b respectively. Comparing the two shows that for $n_\phi = 20$, the absorption is considerably lower and more spread out vertically than for $n_\phi = 10$. This is caused by the wave propagation in the tokamak. For the lower value of n_ϕ , the waves formed a more distinct eigenmode pattern than for higher values. For the case of $n_\phi = 50$, the absorption is very low and not in the center of the plasma anymore, as seen

in Fig. 9c. This is because the wave is not able to reach the plasma center in that case, as discussed in Section V-A.

Similar to AUG, the minority absorption in ITER is lower and more spread out for higher values of n_ϕ . This can be seen by comparing the absorption for $n_\phi = 25$ in Fig. 10a to that for $n_\phi = 70$ in Fig. 10b. For the lower toroidal mode number, an interference pattern can be seen in the absorption. This is an effect of the more prominent ion-ion hybrid layer and the higher amount of wave reflections in that case. However, for both toroidal mode numbers, the absorption is more concentrated to the center of the plasma compared to AUG. This is an effect of the waves in ITER propagating like a beam into the plasma.

The polarization in ITER for the two different plasmas can be seen in Fig. 8, which shows a higher level of left-hand polarization in deuterium-tritium than in deuterium. Since the left-hand polarized component is what resonates with the ions in the plasma, this implies a higher level of absorption in the deuterium-tritium plasma compared to the deuterium plasma. A deuterium-tritium plasma will thus be easier to heat compared to the deuterium plasma when using ICRH.

In terms of heating the plasma through ICRH, ITER will have an advantage over AUG as the absorption will be more concentrated to the middle of the plasma. This is desirable since heating the plasma at the edges will result in a higher power loss to the surroundings.

C. Plasma Impedance

The impedance of the plasma in AUG has large variations for different toroidal mode numbers, as seen in Fig. 13a. The impedance has periodic spikes with intervals of four to five in n_ϕ . The wave reflection in AUG is high enough for an eigenmode pattern to form, which appears to impact the impedance and cause it to oscillate.

In the case of ITER, the impedance plot for the deuterium plasma in Fig. 13b also shows variations for varying toroidal mode numbers, but the effects are much less apparent than for AUG. The deuterium-tritium plasma has no visible oscillations in its impedance plot at all. The amount of reflections was generally lower for deuterium-tritium compared to deuterium, which in turn was lower than for the plasma in AUG, as seen in Tables II and III. As the amount of reflections determines if the waves form an eigenmode pattern or a beam in the plasma, this indicates that the absence of oscillations in impedance for the deuterium-tritium plasma in ITER is because the waves propagate as a beam. For the deuterium plasma, the higher amount of reflections seems to impact the plasma impedance and cause slight oscillations for different toroidal mode numbers.

Since the plotted impedance is proportional to the total absorbed power, as explained in Section III-E, spikes in the impedance plot correspond to values of n_ϕ where there is high power absorption. Aside from the oscillations, the impedance in ITER is of similar size for the two different plasmas, meaning a similar amount of power is absorbed.

The overall shape of the impedance plots for both tokamaks can be seen to be exponentially decreasing for increasing

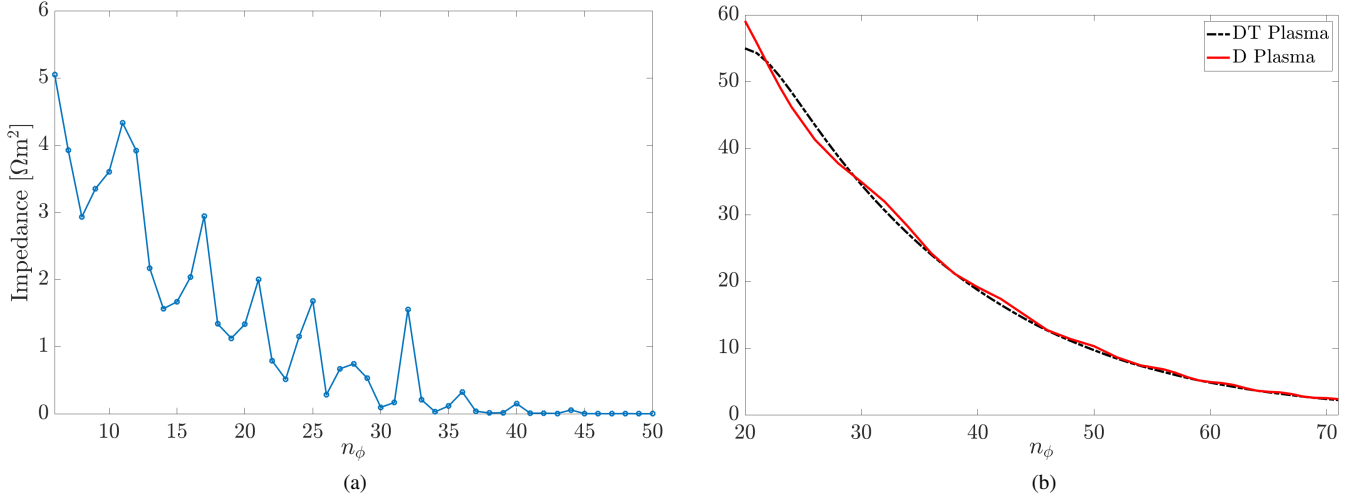


Fig. 13. Plasma impedance for different toroidal mode numbers in (a) AUG and (b) ITER.

values of n_ϕ , indicating that most of the power is absorbed for low toroidal mode numbers. This is as expected from theory, as a higher value of n_ϕ gives an exponentially weaker coupling between the antenna and the plasma, as explained in Section II-F.

D. 3D Fields

The characteristics of the wave propagation and the formation of wave patterns in AUG and ITER, described in Section V-A, are also present in the 3D fields. The eigenmode pattern in AUG, seen in Fig. 11a and 11b, is spread out in the entire tokamak and anti-symmetric about the x -axis due to the antenna current having the same anti-symmetry. The anti-symmetric wave propagation about the x -axis can also be seen in Fig. 12 for ITER, but the waves behave as beams propagating in a narrow field near the antenna.

The ion-ion hybrid layer is clearly visible for both tokamaks in Fig. 11b and 12. In ITER only a small portion of the waves makes it past the ion-ion hybrid layer. As discussed in Section V-A, the hybrid layer is stronger and more apparent for a larger range of n_ϕ in ITER. This causes more wave reflection at the ion-ion hybrid layer in ITER, hence less waves are transmitted through the hybrid layer.

The 3D fields in Fig. 11a, 11b and 12 include all toroidal modes between -50 and 50 in AUG and between -147 and 147 in ITER. According to Fig. 4, the current spectrum continues for a wider range of toroidal mode numbers n_ϕ , than what was included when plotting the 3D fields. To verify that excluding modes higher than 50 in AUG did not have an impact on the obtained result, a 3D field was plotted with modes that ranged between -80 and 80 . All modes above 50 were given the value of the electric field amplitude from the simulation performed for the toroidal mode $n_\phi = 50$. There were no considerable changes in the wave pattern when modes above 50 were included. This indicates that no further simulations with modes ranging beyond ± 50 needed to be performed. The same conclusion can be made when looking

at the weak and fast decaying wave propagation in Fig. 5c, for $n_\phi = 50$ in AUG.

The same verification made for AUG was also done for ITER, but revealed a different result. There were slight differences between the 3D plots that ranged between ± 71 and ± 147 . The 3D field, with modes between -71 and 71 , included "ghost modes". These are wave contributions in places where none should exist. The phenomenon is caused by the missing terms in the Fourier sum in Eq. (22), as it is not practically possible to evaluate for infinite toroidal mode numbers. Most of the ghost modes went away for the 3D field with values of n_ϕ between ± 147 . This indicates that modes beyond ± 71 have a bigger impact on the wave distribution and were therefore not excluded.

VI. CONCLUSION

The goal of the project was to examine what effects the larger size of ITER will have on ICRH. A number of simulations of ICRH in the two tokamaks were made in order to study the electromagnetic wave propagation and impedance in the plasma. Since the electric field can be decomposed into a Fourier series in the toroidal direction, the simulations were made for individual components for different values of the toroidal mode number n_ϕ . The mode numbers included in the simulations were chosen from the current spectrum of the antenna, as it dictates how much power can be delivered to the plasma for different values of n_ϕ . The total electric field could then be visualized in a 3D plot by summing all the simulated components, scaled by the antenna current, for the different toroidal mode numbers.

The results show a clear difference in wave propagation between the two tokamaks. In the AUG simulations, the waves formed a pattern of eigenmodes while simulations of ITER had waves propagating as a beam into the plasma. This is correlated with the amount of wave reflections in the plasma, which was higher in AUG than in ITER. Since the wave was more spread out over a larger volume in AUG, so was the power absorption by the minority. In ITER, where the waves

had the shape of a beam, the minority power absorption was concentrated to the plasma center.

The plasma impedance showed a trend of exponential decrease for increasing toroidal mode numbers in both tokamaks. The impedance graph for AUG had large oscillations for varying n_ϕ . This effect was much smaller in ITER for the deuterium plasma and completely absent for the deuterium-tritium plasma. This could also be tied to the amount of wave reflections for each case. The reflection coefficient for the deuterium-tritium plasma was around a factor ten lower for higher toroidal mode numbers than for the deuterium plasma in ITER, which in turn was around a factor ten lower than for AUG. This indicates an anti-correlation between power absorption and the reflection coefficient, meaning higher absorption gives less reflection.

In terms of heating, it is desirable for the power to be absorbed in the middle of the plasma. Because of this, the results of this project give a clear advantage to ITER, where the simulated waves were able to propagate to the center of the plasma without spreading out due to a high amount of reflections. The coupling between the antenna and the plasma, described by the plasma impedance, is also more robust for ITER than for AUG. Taken together, the results indicate that plasma heating by ICRH will be more efficient in ITER compared to AUG.

ACKNOWLEDGMENT

The authors would like to thank their supervisors Thomas Jonsson and Björn Zaar for their tremendous help during this thesis project. They were always supportive and provided meaningful feedback. They also managed to make a complex field of study feel accessible from the very beginning.

REFERENCES

- [1] M. R. Hannah Ritchie and P. Rosado. (2020) Energy. [Online]. Available: <https://ourworldindata.org/energy>
- [2] V. Masson-Delmotte, P. Zhai, A. Pirani, S. Connors, C. Péan, S. Berger, N. Caud, Y. Chen, L. Goldfarb, M. Gomis, M. Huang, K. Leitzell, E. Lonnoy, J. Matthews, T. Maycock, O. Y. T. Waterfield, R. Yu, and B. Z. (eds.), "Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change," *IPCC*, Aug. 2021.
- [3] H.-O. Pörtner, D. Roberts, M. Tignor, E. Poloczanska, K. Mintenbeck, A. Alegría, M. Craig, S. Langsdorf, S. Löschke, V. Möller, A. Okem, and B. R. (eds.), "Climate Change 2022: Impacts, Adaptation, and Vulnerability. Contribution of Working Group II to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change," *IPCC*, Feb. 2022.
- [4] J. P. Freidberg, *Plasma Physics and Fusion Energy*. Cambridge, UK: Cambridge University Press, 2007, ch. 1.
- [5] —, *Plasma Physics and Fusion Energy*. Cambridge, UK: Cambridge University Press, 2007, ch. 2.
- [6] —, *Plasma Physics and Fusion Energy*. Cambridge, UK: Cambridge University Press, 2007, ch. 3.
- [7] —, *Plasma Physics and Fusion Energy*. Cambridge, UK: Cambridge University Press, 2007, ch. 13.
- [8] 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, KTH, Stockholm, Sweden, 2019.
- [9] J. P. Freidberg, *Plasma Physics and Fusion Energy*. Cambridge, UK: Cambridge University Press, 2007, ch. 15.
- [10] ITER. (2022, Mar.) The iter tokamak. [Online]. Available: <https://www.iter.org/mach>

- [11] Max Planck Institute for Plasma Physics. (2022, Mar.) Introduction: The ASDEX Upgrade tokamak. [Online]. Available: <https://www.ipp.mpg.de/16208/einfuehrung>
- [12] ITER. (2022, Mar.) What will iter do ? [Online]. Available: <https://www.iter.org/sci/Goals>
- [13] S. Li, H. Jiang, Z. Ren, and C. Xu, "Optimal Tracking for a Divergent-Type Parabolic PDE System in Current Profile Control," *Abstract and Applied Analysis*, vol. 2014, p. 2, Jun. 2014.
- [14] P. Vallejos, T. Johnson, R. Ragona, T. Hellsten, and L. Frassinetti, "Effect of poloidal phasing on ion cyclotron resonance heating power absorption," *Nuclear Fusion*, vol. 59, no. 7, p. 076022, Jun. 2019.
- [15] COMSOL® *Multiphysics*. (5.6, build: 401). [Online]. Available: www.comsol.com
- [16] D. K. Cheng, *Field and Wave Electromagnetics*. Harlow, UK: Pearson Education Limited, 2014, ch. 7.
- [17] D. G. Swanson, *Plasma Waves, 2nd Edition*. Bristol, UK: IOP Publishing Ltd, 2003.
- [18] D. K. Cheng, *Field and Wave Electromagnetics*. Harlow, UK: Pearson Education Limited, 2014, ch. 6.
- [19] —, *Field and Wave Electromagnetics*. Harlow, UK: Pearson Education Limited, 2014, ch. 8.
- [20] T. Jonsson, "Waves in fusion plasmas," lecture handouts in course *Electromagnetic Waves in Dispersive Media*, KTH, Stockholm, Sweden, 2019.
- [21] D. K. Cheng, *Field and Wave Electromagnetics*. Harlow, UK: Pearson Education Limited, 2014, ch. 9.
- [22] MATLAB®. (R2020b and R2021b), MathWorks®. [Online]. Available: www.mathworks.com
- [23] V. Bergeaud, L.-G. Eriksson, and D. Start, "ITER relevant ICRF heating scenarios with large ion heating fraction," *Nuclear Fusion*, vol. 40, no. 1, pp. 35–51, Jan. 2000.

CONTEXT K

OBSERVATIONS IN SPACE PHYSICS

POPULAR DESCRIPTION

The secrets of the mysterious plasma hidden in our universe

The arctic air, sharp and cold, fills your lungs with every breath. The sky above you is devoid of light except for the faint glimmer of distant stars. Suddenly, the black void above is filled with vibrant curtains of blue and green light, dancing across the night sky. While you take in their beauty, billions of particles enter Earth's magnetic field to join the dance.

The rare and beautiful light show that can be observed during the night is actually not a show intended for us! It is in fact a phenomenon where particles from space are collected at the poles of the Earth where the invisible magnetic field protecting the planet is the strongest. This happens far, far up in the atmosphere, ten times higher than most commercial airlines fly.

Everyone knows about the three basic states of matter: solid, liquid and gas. In space however, molecules and atoms are heated up so much that they break apart, forming a fourth state of matter, plasma. The secret behind this ionized gas is that it is actually electrically conducting, meaning it will interact with magnetic fields such as those around planets and stars. It is plasma that causes the northern and southern lights and many other phenomena that are, quite literally, out of this world. One example is when particles from Jupiter's moon Io become plasma and are captured by the magnetic field of the planet, glowing all around the moons as auroras. Another example is shock waves in the plasma ejected from the sun when the plasma comes near a planet and its magnetic field, much like how shock waves in air are created around supersonic airplanes.

There are many methods and tools that can be used to study these phenomena. Data can be gathered by sending spacecraft deep into space or by using advanced telescopes like Hubble. This data can then be used to show how the magnetic fields look or calculate how plasma behaves when it collides with them. The extreme conditions of outer space, unlike anything seen on Earth, makes plasma behave in interesting ways. By observing it, we can learn a lot about how the universe works.

SUMMARY OF PROJECT RESULTS

In order to get an understanding of phenomena in space and the underlying physics it is often necessary to analyze data from different satellites and space missions. A concrete example of this is plasma, which is rarely observed on Earth but is commonly found in space. By measuring plasma properties like temperature, pressure, density etc. a greater understanding of the laws of physics in a space plasma environment can be attained. In turn, one can leverage that knowledge to understand more about the conundrum that is outer space. In context K, plasma is the common denominator when analyzing the nature of the magnetic fields surrounding Earth and Jupiter.

In context K, data from NASA's Juno mission and NASA's Magnetospheric Multiscale mission (MMS) were used to study different plasma related properties in the magnetospheres of Jupiter and Earth. The magnetosphere is the cavity around a magnetized planet, which is dominated by its own magnetic field. The Juno probe orbits Jupiter since 2016 and among its many mission objectives is the mapping of the planet's magnetic field using data gathered by fluxgate magnetometers. Launched in 2015, MMS orbits Earth while traveling in and out of the magnetosheath measuring the properties of plasma and the magnetic field. Project group K1 studied the plasma environment surrounding the Jupiter moons Io and Europa whereas project group K2 and K4 studied the bow shock, a collisionless plasma shock wave generated when the solar wind interacts with the magnetic field of celestial bodies, around Jupiter and Earth.

Project group K1 studied Jupiter's magnetosphere, which is loaded with plasma due to the volcanically active moon Io. Significant amounts of material from volcanic eruptions on Io escapes the moon's atmosphere and becomes ionized when it

interacts with the plasma in Jupiter's magnetosphere. The ionized particles get trapped in the magnetosphere and form a torus of plasma that rotates with the magnetic field of Jupiter. Aurorae can be observed on the moons Io and Europa as they collide with the plasma torus and this aurora was assumed to be dependent on the density of plasma at the location of the moon. The project group modeled this plasma torus and compared the model to Hubble Space Telescope observations of brightness of the aurorae. A weak correlation between plasma density and aurora brightness was found. The results from this project are a validation of the method of using one type of measurement data to derive another, in this case luminosity data to model plasma density. In the coming years, the Juno space probe will lower its orbit and take measurements of the Io plasma torus. This new data will create opportunities for future projects to develop a further improved model of the plasma environment of Jupiter.

Project group K2 modeled Jupiter's bow shock and magnetopause by analyzing data from NASA's Juno probe between its arrival at Jupiter in June 2016 and the end of January 2018. The modeling in the project was done by compiling a list containing Juno's position relative to the center of Jupiter for the crossings of either of the aforementioned phenomena as well as the times at which the crossings occurred. The lists were then used to create parabolic models of the phenomena using numerical data regression. By slightly altering the parameters of the function and calculating the distance to all data points from the resulting curve, the most likely position and shape of the bow shock and magnetopause was determined. The final models were compared to previous studies by Huddleston et al. (1998) and Joy et al. (2002) to see how similar they are to one another and discuss the probable causes of any significant differences.

In project K2, data analyzed that covers crossings of Jupiter's bow shock and magnetopause is only localized to a relatively small region in space which complicates the modeling of these phenomena. Launching more space probes to Jupiter would provide more data around Jupiter useful for further studies of the bow shock and magnetopause. The Jupiter Icy Moons Explorer and Europa Clipper probes are planned to be launched in April 2023 and October 2024 respectively and will serve as great sources of new planetary data due to their close proximity to Jupiter. This data could be used in further studies to build upon the models of Jupiter's bow shock and magnetopause created in the present project.

In project K4, the change in entropy for electrons crossing Earth's bow shock was studied using satellite data from MMS. This project is a continuation of the paper by M. Lindberg et al. (April 2022) , where it was deduced that a low electron plasma beta, the plasma pressure divided by the magnetic pressure, indicated a large change in entropy. By designing an algorithm using a database of identified shock crossings, the group was able to sort and filter the large amount of data according to the following properties; plasma beta, the angle of incidence and date. The entropy change was then related to different plasma parameters such as Alfvén Mach number, Whistler Mach number, solar wind temperature, ion ram pressure and electron number density. As a result, we could see a strong dependence on the Alfvén Mach number and the solar wind temperature for the entropy change in electrons traveling with the solar wind into Earth's magnetic field.

In further studies similar to project K4, it will be important to develop a more precise algorithm for the correction of the MMS data disturbances to get more reliable results. It would also be an interesting project to look into the math behind a better model for correcting these disturbances.

IMPACT ON SOCIETY AND ENVIRONMENT

An ethical reflection about space research is not straightforward as new results in space plasma physics have no direct impact on society as a whole or on individuals, as compared to other research areas such as cancer medicine or sustainable energy sources. In addition to this, constructing the satellites, probes and space telescopes necessary for such research and launching them into space requires a large amount of economic resources. On one hand, one could argue that it would be more ethical to allocate these resources to research with more immediate practical applications. A sizable portion of space research is funded by taxpayer money which raises the question of whether it would be more ethical to spend that money on welfare programs or infrastructure. This would directly help combat social injustices and improve quality of life in general rather than merely help humans better understand the universe.

On the other hand, since all science is predicated on fundamental research being done, any of the discoveries made in this field may prove useful for practical applications in the future. For example, studying space phenomena might help people understand similar terrestrial phenomena better and can be used to predict solar storms. Using these predictions, potential damage caused by such storms inducing high currents in the electrical grid could be avoided by, for example, disconnecting the grid before a storm hit Earth. With all this in mind we are convinced that spending resources on observations in space physics, rather than other expenditures, is motivated.

In order to describe the impact on individual people from this type of basic research with the sole purpose to gain understanding of our world, we must look at the benefits from new knowledge about the universe for education and for inspiration in general. By gaining knowledge of outer space and its properties, we are constantly updating the information being taught in education at different levels. This in turn will have a positive effect on interest in science as a whole. The never-ending search for understanding and knowledge can be seen as a part of human nature which means space research will never lose its purpose.

The field of space research does not contribute to making human civilization more sustainable as other fields such as automation or electrical power engineering. In order to gather the data necessary to perform such research often space probes must be launched into space. This is in most cases done through rocket launches, which require the combustion of a sizable amount of fuel that results in the emission of greenhouse gasses and other harmful compounds. While these are relatively insignificant compared to the emissions from regular forms of transport and power plants, it is not negligible, especially as launches become cheaper and, as a result, more common.

Another problem is that attempted space launches run the risk of failure. This can produce large amounts of debris and potentially spread dangerous substances carried by the spacecraft in a wide area in addition to effectively wasting all resources that were expended on the attempt. Even successful launches can result in debris on the ground, the sea and in orbit due to discarded parts. However, measures are taken to mitigate all these problems. Rocket companies such as SpaceX, Rocket Lab and Blue Origin could help reduce the amount of spent resources by reusing large parts of the rocket for multiple launches. It is possible to use spacecraft and satellites equipped with nets, claws or other tools to capture and clean Earth's orbit from space debris to lower the risk of collisions.

Transitioning to more effective solar-power technologies rather than the nuclear-power in spacecraft used in the exploration of the outer planets such as Jupiter could also reduce the risks associated with a possible launch failure. Finally, transitioning to cleaner rocket fuels such as methane can reduce the environmental impact compared to using more traditional rocket fuels. Therefore the current issues of sustainability in the field of space research are not only theoretically solvable, but also in the process of being solved. In summary, there are no significant arguments against but many arguments in favor for further scientific satellite missions and space research in general.

Using Jupiter's Moon Io as a Plasma Probe

Erik Hedenström and Anton Petré

Abstract—The structure of the plasma in Jupiter's vast magnetosphere is complicated and not fully understood. One way to study the plasma is to look at auroral emissions from the moon Io as it moves through different regions of the plasma torus that surrounds Jupiter. In this paper, the correlation between aurora brightness on Io and the plasma density at the position of the moon is investigated. If a correlation exists, auroral emissions on Io could be used as a diagnostic for the current state of Jupiter's plasma environment. For this purpose, a model of the Io plasma torus is developed, combining ideas from different existing models. The model is compared with observations of aurorae on Io made by the Hubble Space Telescope. Io's position at the time of the observations is obtained with SPICE, a software developed by NASA. A moderate correlation is found when using the whole data set of observations. However, a strong correlation is found for observations on the dusk side of Jupiter. Strong correlations are also found when studying individual years and epochs.

Sammanfattning—Strukturen på plasman i Jupiters vidsträckta magnetosfär är komplicerad och inte fullständigt känd. Ett sätt att studera plasman är att undersöka ljuset från polarsken på månen Io då den passerar genom olika regioner av det torusformade plasmamolnet som omsluter Jupiter. I denna artikel undersöks korrelationen mellan polarskenets ljusstyrka och plasmans densitet kring månens position. Om ett sådant samband finns skulle ljusstyrkan hos månens polarsken kunna användas som diagnostik för plasmans aktuella tillstånd. För detta ändamål utvecklas en modell av plasmatorusen genom att kombinera idéer från flera tidigare modeller. Modellen jämförs sedan med observationer av polarskenet på Io genomförda med rymdteleskopet Hubble. Månens position vid de olika tidpunkterna bestäms med hjälp av SPICE, en mjukvara utvecklad av NASA. En måttligt stark korrelation uppnås när hela datamängden används. När däremot endast data från Jupiters gryningssida används uppnås en stark korrelation. Det hittas även starka samband när enskilda år studeras.

Index Terms—aurora, Hubble Space Telescope, Io, Jupiter, magnetosphere, plasma.

Supervisor: Lorenz Roth

TRITA number: TRITA-EECS-EX-2022:154

I. INTRODUCTION

The Galilean moons of Jupiter are immersed in the large amount of plasma contained in the planet's vast magnetosphere [1]. The main source of plasma is volcanic activity on Io, the innermost Galilean moon, releasing large amounts of SO_2 . Some of this material escapes the moon's atmosphere and gets ionized when it interacts with the plasma in Jupiter's magnetosphere. The ionized particles are trapped by the magnetosphere of Jupiter and forms a torus around the planet that corotates with the planet's magnetic field [2]. This torus is called the Io plasma torus and is the focus of interest in this paper since this is the region of the magnetosphere Io moves through. An illustration of the plasma torus is shown in Fig. 1.

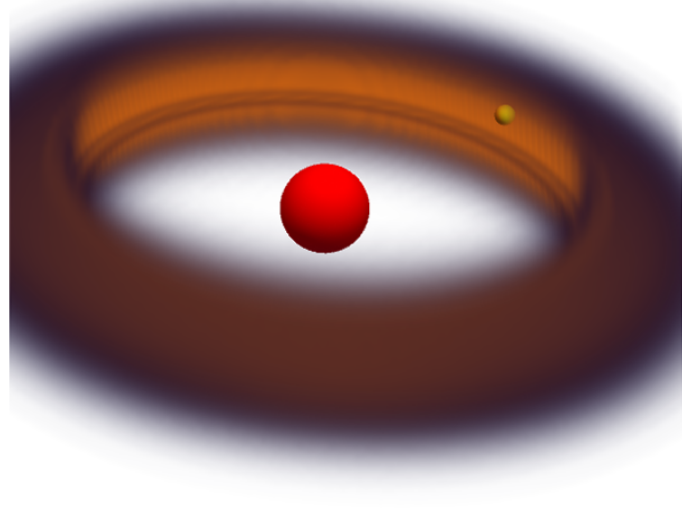


Fig. 1. The Io plasma torus surrounding Jupiter (red) with Io (yellow). Darker color and higher transparency means lower plasma density.

Another consequence of the interaction between the plasma and the atmosphere of Io is excitation of particles, which give rise to aurora [3]. The aurora is mainly a consequence of electrons in the plasma torus colliding with oxygen and sulfur atoms. It is believed that the auroral emissions depend on the density of electrons, hence the density of plasma [3]. These auroral emissions have been observed by several telescopes and spacecraft since the discovery of the plasma torus in 1976 [4]. In this paper, data from the Hubble Space Telescope is used to obtain the brightness of aurorae at 66 different occasions. These brightnesses are average values from images such as the one in Fig. 2 and are given in kiloRayleigh (kR).

To investigate the correlation between plasma density and aurora brightness, a model of the plasma torus is needed. Many attempts to create a model and make it agree with observational data have been made before. However, the plasma torus is still not completely understood and it has proven difficult to create a model that works for different observing epochs [4]. In this paper, a model is created by combining different models for the purpose of studying how the modeled plasma density impacts the brightness of aurorae. Emphasis is placed on how different parameters in the model affect the correlation between plasma density and aurora brightness. As a measurement of the correlation, the Pearson correlation coefficient (ρ) is computed.

The objective of this project is to find a correlation between plasma density and aurora brightness. Finding a strong correlation would enable the use of brightness data in addition to

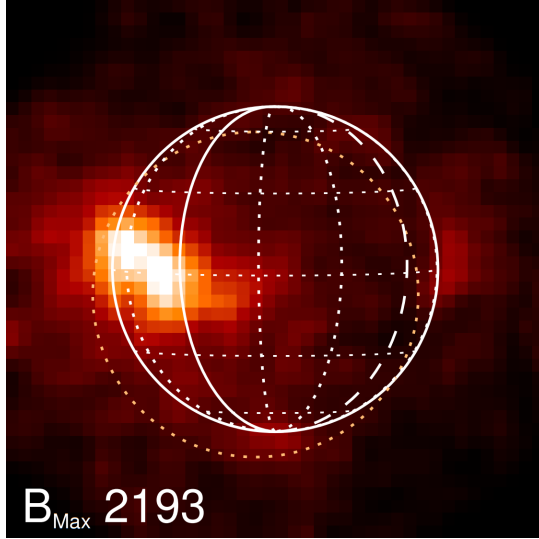


Fig. 2. An image of aurora on Io taken with the Hubble Space Telescope (1998). B_{\max} is the brightness in Rayleigh of the brightest pixel.

density measurements when observing the plasma environment around Jupiter, increasing the amount of possible observations.

II. METHOD

A. Model

As a starting point, two models were combined to create a model of the torus. The first model describes the shape of the centrifugal equator, the farthest point on each field line from Jupiter's spin axis. A common approximation is to assume that the centrifugal equator is confined to a flat, tilted plane. This is what is obtained if the magnetic field is assumed to be a perfect dipole field. However, different tilt angles are used in different papers. In this paper, a tilt angle of 6.4° is used, as suggested by [4]. A more accurate description of the centrifugal equator is presented in [5], which takes into account the fact that electric currents are present in the magnetosphere of Jupiter which affect the shape of the magnetic field. The tilt angle of the centrifugal equator as function of the distance from Jupiter is described in [5] as

$$\theta(r, \varphi) = [a \cdot \tan(h(b \cdot r - c)) + d] \sin(\varphi - e), \quad (1)$$

where r is the distance from the center of Jupiter measured in Jupiter radii and constants a - e are given in appendix A. The value used for the Jupiter radius (R_j) in this paper is 71,492 km. This description of the centrifugal equator is later referred to as the curved model of the centrifugal equator, due to the slightly curved shape. Both the flat model and the curved model of the centrifugal equator are investigated in this paper.

The second model used to describe the plasma torus is a model of the plasma density as function of the distance from the plane of the centrifugal equator and the distance from Jupiter [6]. The torus is divided into four parts: the cold torus, the ribbon, the warm torus and the extended torus. The extended torus is used to describe the torus for large distances

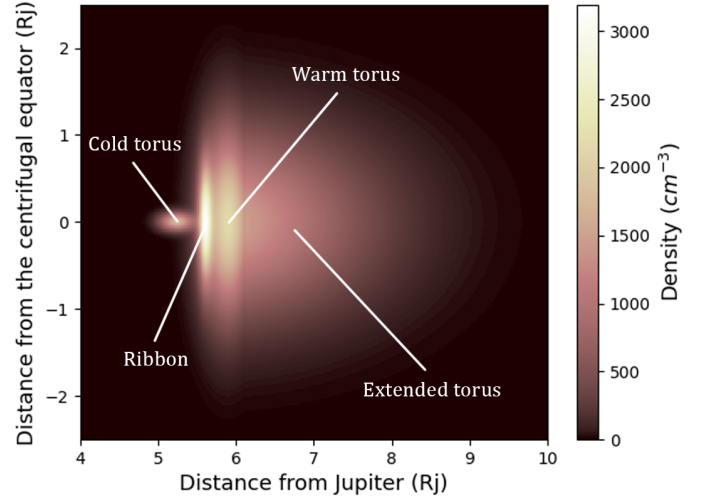


Fig. 3. Plasma density as function of distance from the center of Jupiter and distance from the plane of the centrifugal equator according to Eq. 2.

from Jupiter. The expression derived in [6] for the plasma density is written as

$$\begin{aligned} N(r < 6.1, h) &= N_1 e^{-\frac{(r-C_1)^2}{(W_1)^2}} e^{-\frac{h^2}{(H_1)^2}} + \\ &N_2 e^{-\frac{(r-C_2)^2}{(W_2)^2}} e^{-\frac{h^2}{(H_2)^2}} + N_3 e^{-\frac{(r-C_3)^2}{(W_3)^2}} e^{-\frac{h^2}{(H_3)^2}}, \quad (2) \\ N(r > 6.1, h) &= N_4 e^{-\frac{(r-C_4)^2}{(W_4)^2}} e^{-\frac{h^2}{(H_4)^2}}, \end{aligned}$$

where r is the distance from the center of Jupiter and h is the distance from the plane of the centrifugal equator. N_1 , N_2 , N_3 and N_4 are the scale peak densities of the cold torus, the ribbon, the warm torus and the extended torus respectively. C_1 , C_2 , C_3 and C_4 are the central positions of the different regions. W_1 , W_2 , W_3 and W_4 are the radial widths and H_1 , H_2 , H_3 and H_4 are the scale heights. Values for the constants are provided in appendix A and in [6]. The model described by Eq. 2 is displayed visually in Fig. 3. Several modifications were made to this model. One assumption in the model is that the plasma torus extends vertically from the plane of the centrifugal equator. However, it is reasonable to believe that the plasma torus actually extends along the magnetic field lines of Jupiter. This effect was added to the model by modifying Eq. 2 to extend along the field lines of a dipole located in the center of Jupiter. The formula for this distance is derived in appendix B and the modified torus is displayed in Fig. 4.

Another assumption made in the base model is rotational symmetry around Jupiter. However, according to several sources, there exists a dawn-dusk asymmetry, with the plasma torus being closer to Jupiter on the dusk side of Jupiter. According to [7], the average position of the peak density of the ribbon (the C_2 -parameter) is located at about 5.56 Jupiter radii distance on the dusk side and 5.84 on the dawn side, adjusted to the Jupiter radius used in this paper. This effect was added to the model in two different ways. In one way by assuming that the torus is elliptic and in another way by assuming that the torus is still circular, but shifted slightly in the direction of the dawn side. This can be accomplished by

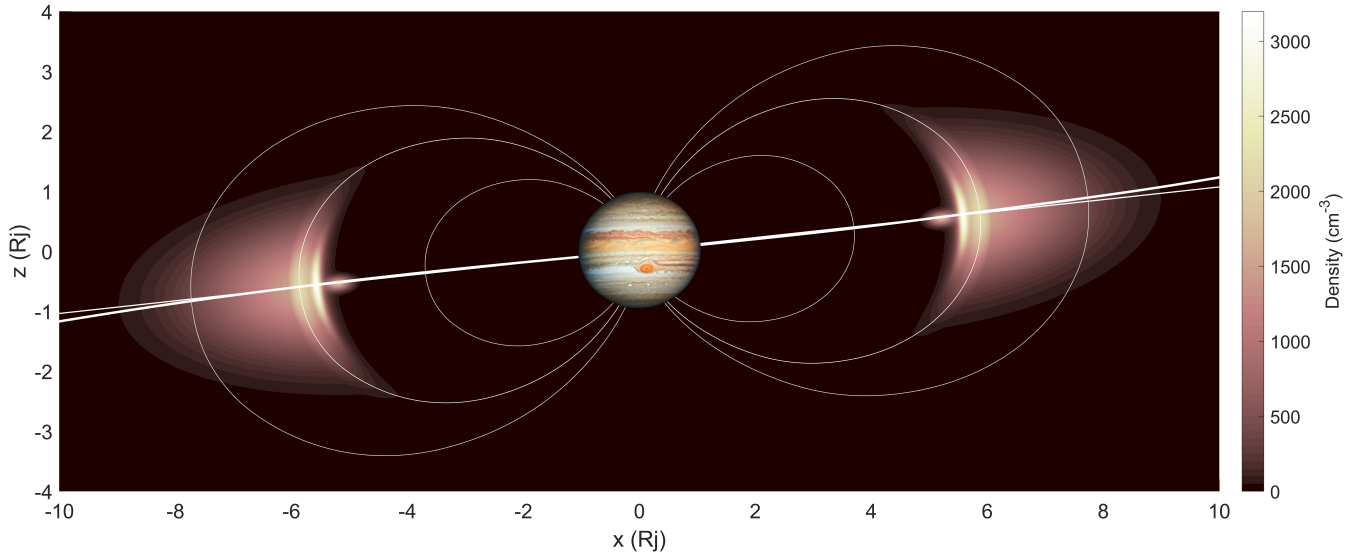


Fig. 4. The plasma torus plotted in the xz -plane in jovicentric coordinates, where the z -axis corresponds to Jupiter's spin axis. The thick white line represents the curved centrifugal equator described by Eq. 1. The thin line is the flat centrifugal equator with a 6.4° tilt. The plasma density is computed with the modified version of Eq. 2 that bends the torus along the dipole field lines as illustrated. The image of Jupiter is cut out from an image by NASA, ESA, A. Simon (Goddard Space Flight Center) and M.H. Wong (University of California, Berkeley) taken with Hubble 2019.

making the C -parameters in Eq. 2 follow either an elliptic function or a sinusoidal function, with minimum on the dusk side and maximum on the dawn side. In this paper, only the values of the C_2 -parameter are reported, but all other C -parameters are always adjusted exactly the same way as the C_2 -parameter. In other words, the distances between the different regions of the torus are always maintained constant.

Lastly, one effect discussed by [4] is that the plasma torus actually deviates about 20 degrees from the true dawn-dusk direction. Different angles of deviation from the true dawn-dusk direction and different combinations of the modifications of the model are investigated in this paper.

To compare the plasma density with the aurora brightness, it was necessary to determine the position of Io at the times of the aurora observations. This was done through SPICE. SPICE is a tool developed by NASA and can calculate the data needed for the calculations in this paper [8]. To use SPICE however, a coordinate system had to be chosen. It was chosen to use the Jupiter System III coordinate system which is described in detail in [9]. It fits this project well since it follows the movement of Jupiter, both through space and in rotation. That preserves coordinates on Jupiter and the plasma torus, simplifying calculations.

B. Linear regression

In this project, linear regression will be used to investigate the dependency of the brightness of the aurorae on the density of plasma. These linear regressions will be represented in the form

$$y = mx + c. \quad (3)$$

As a measurement of the strength of the correlation, the Pearson correlation coefficient will be used. It is a way to

compare fits, even fits to different data sets, and determine which fit is better. The Pearson coefficient ranges in value from -1 to 1 where -1 indicates perfect negative correlation, 0 no correlation and 1 perfect correlation as described in [10]. A commonly used range of values for different strength of correlation is weak correlation between 0 and 0.3, moderate correlation between 0.3 and 0.5 and strong correlation for Pearson coefficients higher than 0.5, with ranges also given in [10]. The criteria for using the Pearson coefficient are fulfilled in our data set since both aurora brightness and plasma density are measured on continuous scales and the data points are considered independent. However, some data points are closer in time than others, as can be seen in Appendix C, and could therefore be argued to depend on each other. On the other hand, most of the data points are quite spread out in both time and space, which should make the Pearson coefficient a quite good measurement of the correlation. Nonetheless, the Pearson coefficient does not measure causation and must therefore be used with caution.

III. RESULTS

The aurora brightness data used in this project consists of 66 observations made with the Hubble Space Telescope between the years 1997 and 2019. The frequency of different brightness levels is plotted in Fig. 5.

Before a complete model of the plasma torus was developed, it was investigated if aurora brightness on Io and the moon's distance from the centrifugal equator were correlated. To study the relation, a least squares fit was made between the two quantities and the Pearson correlation coefficient was computed. The result is presented in Table I, where "Distance, flat cent. equ." represents the flat centrifugal equator with a 6.4° tilt and "Distance curved cent. equ." represents the curved centrifugal equator described by Eq. 1. The latter can also be seen in Fig. 6.

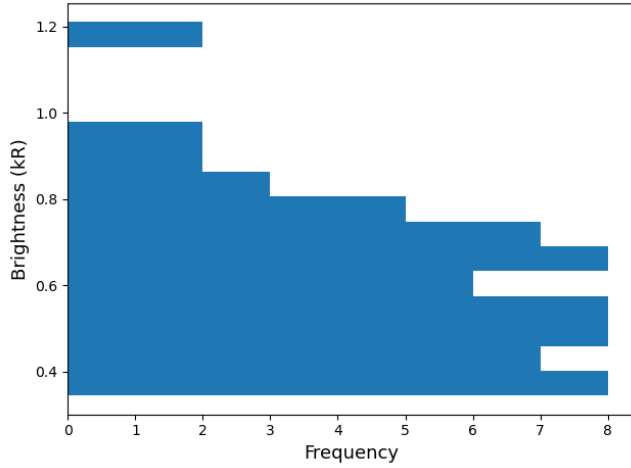


Fig. 5. The frequency of different brightness levels in the studied data set.

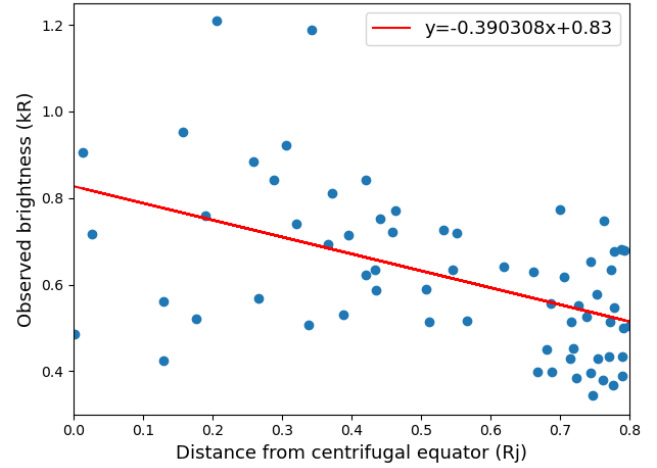


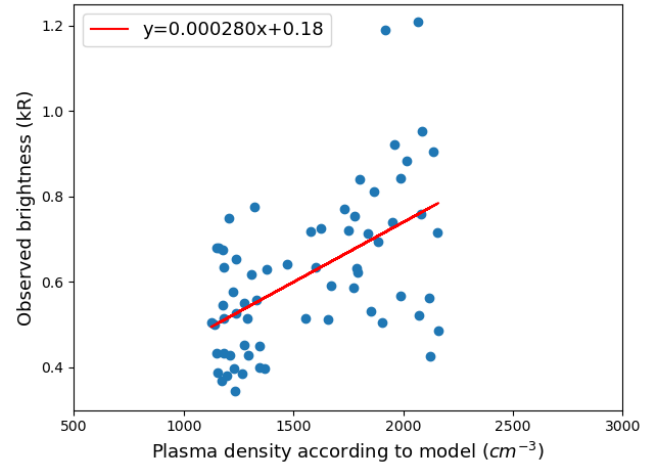
Fig. 6. Brightness data as a function of the absolute distance from a curved centrifugal equator.

TABLE I
PEARSON COEFFICIENTS AND LINEAR FITS FOR CORRELATIONS

Description	Fig.	m (10^{-4})	c	ρ
Distance, flat cent. equ.	-	-4709.13	0.83	-0.504
Distance curved cent. equ.	6	-3903	0.83	-0.504
Base model	7	2.80	0.18	0.540
Circular torus, dawn-dusk 1	8	2.09	0.31	0.451
Circular torus, dawn-dusk 2	9	3.02	0.2	0.567
Elliptical torus, dawn-dusk 1	-	2.09	0.31	0.451
Elliptical torus, dawn-dusk 2	-	3.02	0.2	0.566
Field line model	10	2.4	0.26	0.554
Shifted model 1	-	2.1	0.32	0.402
Shifted model 2	-	3.34	0.18	0.556
Shifted model 3	11	3.84	0.19	0.642
Full model	12	3.69	0.21	0.63
Year 1997 base model	13	3.56	0.13	0.988
Year 1998 base model	-	6.24	-0.23	0.814
Year 2000 base model	13	3.78	-0.03	0.891
Year 2013 base model	13	4.33	-0.04	0.897
Year 2014 base model	-	1.60	0.24	0.634
Year 1997 full model	14	5.39	0.16	0.67
Year 2000 full model	14	4.76	-0.01	0.734
Year 2013 full model	14	6.47	-0.02	0.949
Only dawn base model	15	0.49	0.57	0.157
Only dusk base model	15	3.81	0.01	0.656
Only dawn full model	-	1.23	0.49	0.226
Only dusk full model	-	4.45	0.15	0.7
Model optimized for dusk	16	3.51	0.22	0.806

The first model of the plasma torus used in this paper is a combination of Eq. 1 and Eq. 2. This is referred to as the "base model". With this model, it was possible to start investigating the relationship between aurora brightness and plasma density. A least squares fit was made and the correlation coefficient was computed. The result can be seen in Table I and in Fig. 7.

Several modifications of the base model were made. The first modification was to add a dawn-dusk dependence in the two different ways described in section II. Two different sets of values for the ribbon center position constant (C_2) were used. The first combination is $C_{2,min} = 5.56$ for the dusk side and $C_{2,max} = 5.84$ for the dawn side, which are the mean values of the observed position of the ribbon at the two sides of Jupiter according to [7]. The second combination was found numerically in an attempt to maximize the correlation

Fig. 7. Brightness data plotted against plasma density according to the *Base model* of the plasma torus.

coefficient (ρ). The strongest correlation was achieved with $C_{2,min} = 5.47$ and $C_{2,max} = 5.74$. The results are presented in Table I, where "dawn-dusk 1" represents the first set of C_2 -values and "dawn-dusk 2" represents the second set. Plots for the circular implementation of the dawn-dusk dependence are presented in Fig. 8 and in Fig. 9.

The second modification of the base model was to add a dipole field to make the torus extend along the field lines, as described in section II. This is referred to as the "Field line model" in Table I. The result is also presented in Fig. 10 and the new shape of the torus can be seen in Fig. 4.

The third modification of the base model was to rotate the torus to study the effect of the proposed deviation from the true dawn-dusk direction described in section II. Since the base model is circular, this modification cannot be studied without the dawn-dusk-modification. The two different sets of C_2 -values were used together with the circular version of the dawn-dusk dependence and a 20-degree rotation, as proposed by [4]. The results are presented as "Shifted model

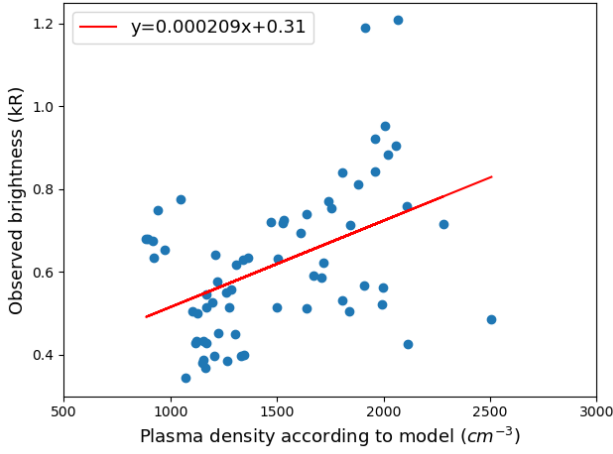


Fig. 8. The *Circular torus, dawn-dusk 1* model, with $C_{2,min} = 5.56$ on the dusk side and $C_{2,max} = 5.84$ on the dawn side, as suggested by [7].

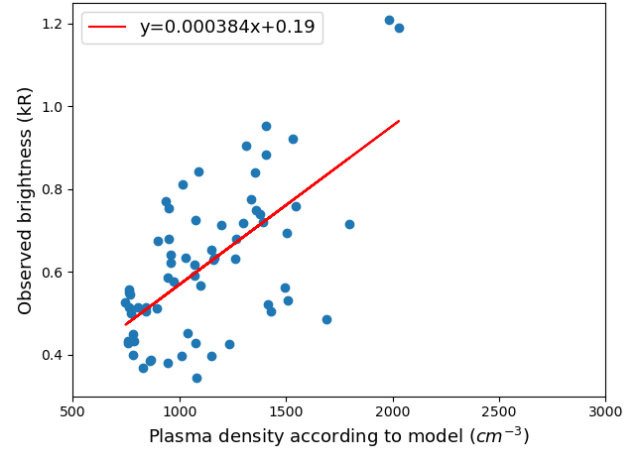


Fig. 11. The *Shifted model 3* of the plasma torus with 20° shift, optimized for lowest ρ .

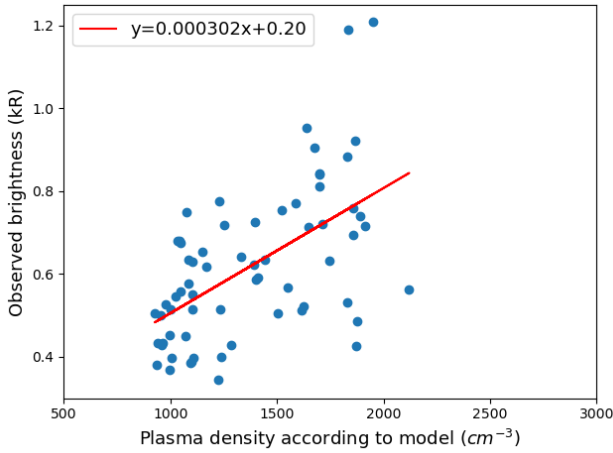


Fig. 9. The *Circular torus, dawn-dusk 2* model, with $C_{2,min} = 5.47$ on the dusk side and $C_{2,max} = 5.74$ on the dawn side, optimized for lowest ρ .

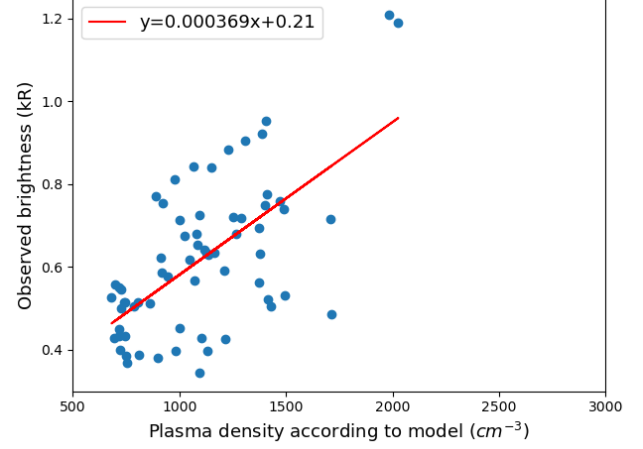


Fig. 12. The *Full model* of the plasma torus.

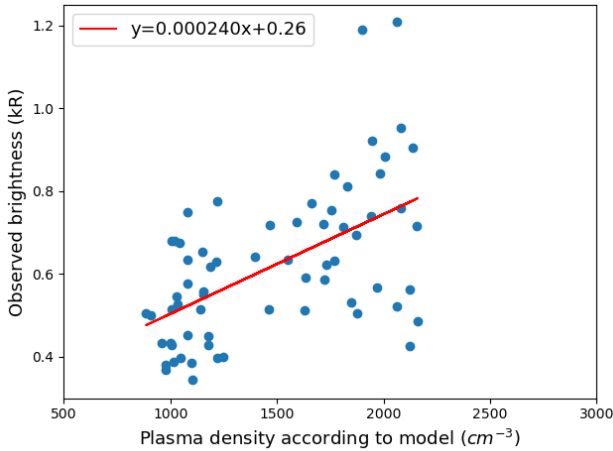


Fig. 10. The *Field line model* plasma torus.

1" and "Shifted model 2" in Table I. In an attempt to improve the correlation coefficient, a third model was created by

optimizing the C_2 -values for different rotation angles. It was found that a rotation angle of 20° with $C_{2,min} = 5.33$ and $C_{2,max} = 5.81$ gave the best correlation. This is referred to as "Shifted model 3" in Table I and the result is also displayed in Fig. 11. Another model was created by adding the field line modification to "Shifted model 3". The result is displayed in Fig. 12 and is referred to as the "Full model" in Table I since it combines all additions to the base model. This model can also be seen in 3D in Fig. 1.

No more modifications to the base model were made. However, several of the above mentioned modifications were studied closer with subsets of the aurora observations. First, observations from individual years were studied separately together with the base model. The years 1997, 1998, 2000, 2013 and 2014 were considered. The results are presented in Table I. A plot with linear fits for the years 1997, 2000 and 2013 is displayed in Fig. 13. The same years were also studied with the "Full model" and the results are presented in Table I and the linear fits are displayed in Fig. 14.

Lastly, the dawn side and the dusk side of Jupiter were

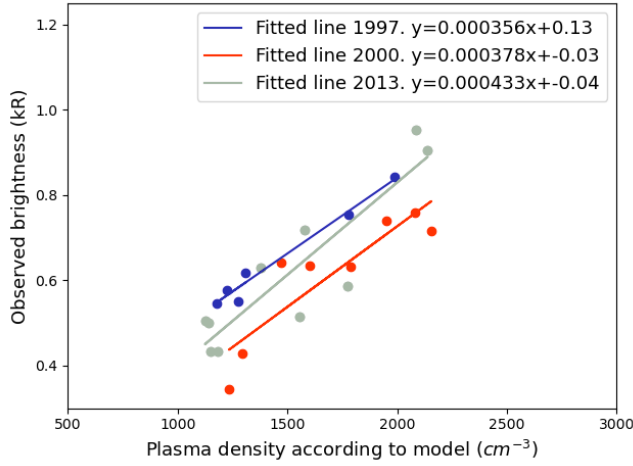


Fig. 13. The *Base model* with linear regression for 1997, 2000 and 2013 individually.

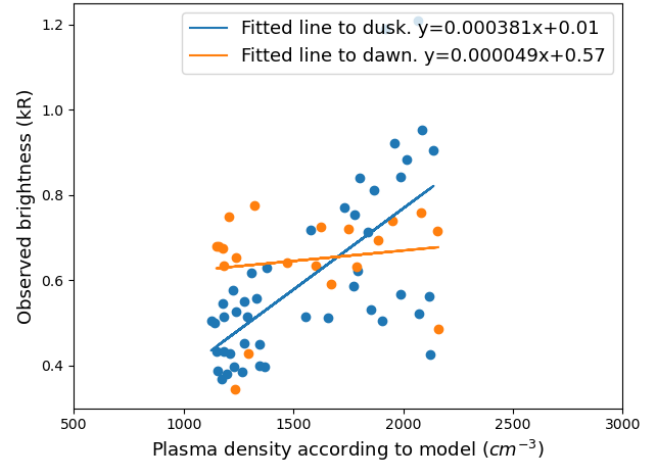


Fig. 15. The *Base model* divided by coordinates on the dusk and dawn side of Jupiter.

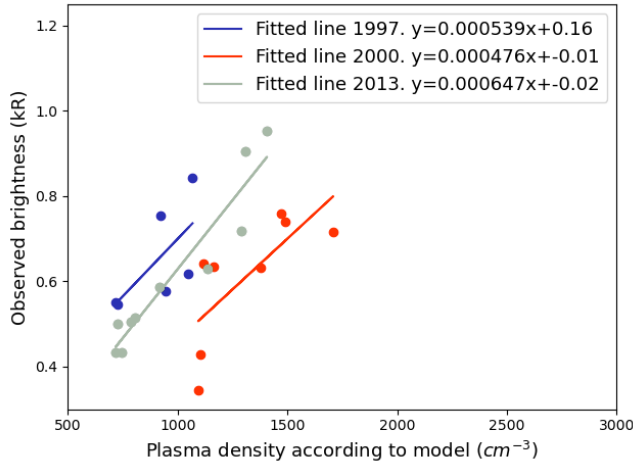


Fig. 14. The *Full model* with linear regression for 1997, 2000 and 2013 individually.

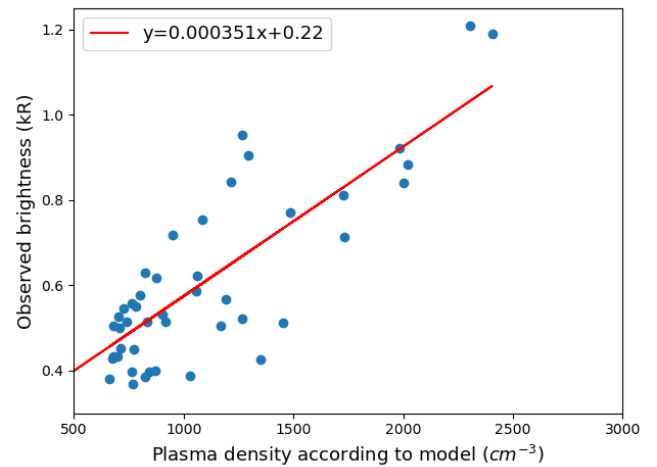


Fig. 16. *Model optimized for dusk*, plotting only the dusk side coordinates.

studied separately. Two least squares fits are displayed in Fig. 15 - one for aurora observations on the dawn side and another one for the dusk side. The results are also presented in Table I together with the results obtained with the "Full model" instead of the base model. Another model was created in an attempt to optimize the correlation for the dusk side. This model is referred to as "Model optimized for dusk" in Table I and was achieved by setting $C_{2,min} = 5.12$ and $C_{2,max} = 8.19$ with a rotation of 0° . The linear fit is displayed in Fig. 16. Several attempts were made to create a similar model optimized for the dawn side, but no significant improvement of the correlation was found. Therefore, it was decided to not present these results.

IV. DISCUSSION

A. Linear correlation

The first result to discuss is the correlation between aurora brightness on Io and the moon's distance from the centrifugal equator. Some kind of correlation seems to exist, however, it

is not very strong. It is marginally stronger for the curved centrifugal equator than the flat, but the difference is so small that it cannot be concluded which one is better. Nonetheless, it was decided to use the curved centrifugal equator in the base model in this paper since it theoretically should be a better description of reality than the flat version.

A stronger correlation is found between aurora brightness and plasma density, as can be seen in Fig. 7. However, with a Pearson coefficient of 0.540, it is still not a very strong correlation. Different modifications were made to the base model to study if the correlation could be improved by improving the model of the plasma torus. The dawn-dusk modification of the base model did not work very well with the C_2 -values suggested by [7]. This can potentially be a consequence of studying different epochs or using different models for the plasma torus. However, a strong correlation was found by modifying the C_2 -values to optimize the correlation. On the other hand, it is hard to say if this improvement of the correlation is due to an improvement of the plasma model or just a lucky coincidence that some data points happen to line

up perfectly for this particular set of C_2 -values. Nonetheless, it is a promising result.

When comparing the two different implementations of the dawn-dusk dependence described in section II, no significant difference was found. That is not a very surprising result since the difference between the two implementations is very small. It was decided to use the circular implementation as standard option.

The magnetic-field-line modification of the base model resulted in a decrease in the slope in the linear fit since the lowest densities moved to even lower densities. However, it did not improve the correlation coefficient more than marginally. Even if the correlation may be considered strong in some sense, it can still be difficult to make a good linear fit with a correlation coefficient of that magnitude. For some configurations of the different modifications to the base model, adding the the magnetic-field-line modification actually resulted in a slightly lower correlation coefficient. Therefore, it cannot be concluded that the magnetic-field-line modification was neither an improvement nor a deterioration of the model. This is probably due to the fact that the constants in Eq. 2 were determined by [6] using the vertical model of the plasma torus. To be able to further study the significance of the magnetic-field-line modification, these constants would need to be re-determined for the modified model.

The implementation of the 20° -deviation from the true dawn-dusk direction suggested by [4] did not seem to improve the "dawn-dusk 1"-model. However, it has already been concluded that the C_2 -values used in that model did not work very well. Therefore, this result may not be very interesting. When the same deviation was added to the "dawn-dusk 2"-model, using the second set of C_2 -values, nothing significant happened either. However, the most remarkable result may be what is described as "Shifted model 3" in Table I. With $C_{2,min} = 5.33$ and $C_{2,max} = 6.08$ and a deviation of 20 degrees, a correlation coefficient of 0.642 was achieved, which is the strongest correlation achieved with the whole data set in any of the many modifications to the base model. In this model, the two data points with the highest brightness values also have the highest densities. This may indicate that these two points are possibly not outliers, but instead correspond to aurora observations in a region of very high plasma density. Another interesting aspect is that no other deviation angle than 20 degrees produced a stronger correlation with any configuration of C_2 -values when the whole data set was used. According to [4], the deviation from the true dawn-dusk direction should be about 20 degrees. It cannot be ruled out that this is just a coincidence due to the limited amount of data. Nevertheless, a deviation of 20 degrees seems to work very well with this particular set of aurora observations.

When considering data from different years individually, the correlation improved significantly. Firstly, it is worth noting that 1998 is the year that contains the two outliers in the brightness data, so any conclusions that use that year should be treated with caution and that year will not be considered in the further discussion here. Data points from 1997 proved to line up almost perfectly, resulting in a correlation coefficient of 0.988 for the base model. Overall, some years were better than

others, but in general, the correlation improved compared to using the full data set. This could be due to each series having fewer points which leads to a more linear structure, but two points speak against this. The first point is that the different years have similar m as seen listed in Table I and in the similar inclination of graphs in Fig. 13 and Fig. 14. This points towards the model being descriptive over the shorter time intervals separately but needing some sort of time adjustment that models the change in c for the different years. Another point for the years being correct individually is the fact that no year, not even 1998, gives rise to horrible results. If the linearity was a product of chance it would be expected that the years would vary in quality, but all of the measured years are as good as the data set as a whole or better. However, due to the small number of years with an adequate number of measurements it is not possible to draw any final conclusions from these observations.

When dividing the data set into two parts, one for data on the dusk side of Jupiter and another for data on the dawn side, it proved that the correlation was significantly better for data on the dusk side. This was evident already when using the base model, where in practice no correlation was found on the dawn side, whereas a strong correlation was found on the dusk side. Even when trying different rotations of the torus and different C_2 -values, no significant correlation for observations on the dawn side was found. For some configurations, there even appeared to exist a negative correlation on the dawn side, which makes no physical sense. The poor correlation for data on the dawn side makes it difficult to optimize a model for the whole data set. It is possible that the parameters used to create "Shifted model 3" managed to reduce the impact of the dawn side data by making the densities agree better with the dusk side data, hence a strong correlation for the whole data set was achieved. Because of the significant difference in correlation between dawn side data and dusk side data, it might be better to study the two data sets separately.

The "Model optimized for dusk" was developed to investigate how strong correlation could be achieved by only considering the dusk side data. Considering the relatively large amount of data, a correlation coefficient of 0.806 is a strong indication of the existence of a correlation. However, the most remarkable result may not be the correlation coefficient itself, but the parameters used to obtain it. It was found that setting $C_{2,min} = 5.12$ on the dusk side and $C_{2,max} = 8.19$ on the dawn side yielded the strongest correlation. These values are completely out of range of the values provided by [7] for the position of the ribbon. Most remarkably, using these values significantly increases the radius of the torus. This results in a considerable increase of the distance from Jupiter to the ribbon on the day and night sides of the planet. It was found that increasing the distance on the day and night sides, the correlation on the dusk side of Jupiter improved. However, further studies are necessary to investigate if this has any physical meaning or if it is just a coincidence.

B. Limitations of the approach

As described in the introduction of this paper, a somewhat simplified model of the plasma torus has been used. Some

effects that are included in other, more advanced models have been ignored for simplicity. Two examples of such effects are temperature and plasma drift, which are included in the model developed by [4]. Also, the dawn-dusk asymmetry was implemented in a simplified way in this paper. According to [7], the position of the ribbon not only varies between dawn and dusk, but also varies periodically with the longitude of Jupiter. This periodic behaviour was ignored for simplicity. Another approximation used in this paper and by various sources to this paper is that Jupiter has a dipole magnetic field. It is possible that different results would be achieved with a model that implements the true magnetic field of Jupiter, although this may not be the most important aspect regarding accuracy. Because of these simplifications, it is possible that the correlation between plasma density and aurora brightness is stronger in reality than shown in this paper.

Another inaccuracy comes from altering the base model of the plasma torus as that potentially makes it deviate even more from the data used to calculate the parameters in the base model in the first place. To get a more accurate model, all parameters would need to be adjusted for every change in the model to make the model agree better with the original observational data of the plasma. This would also provide a second data set for the model to adhere to, preventing over fitting to the brightness data.

Another effect to consider is that the plasma density in the vicinity of Io is most likely affected by Io itself. Consequently, the actual value of the plasma density may differ from the value used in the comparison with aurora brightness. However, the focus of this project was to study the correlation between the unperturbed plasma density and the aurora brightness. This is interesting to study since it is reasonable to believe that the plasma density in the vicinity of Io is related to the plasma density outside this region. Hence, if the aurora brightness were related to the plasma density, a higher brightness should be observed as Io moves through a region of the torus with higher plasma density. Although, this relationship may not necessarily be linear. However, due to the strong linear correlation found on the dusk side between the unperturbed plasma density in the torus and the aurora brightness on Io, it is reasonable to believe that at least some kind of linear correlation exists.

V. CONCLUSIONS

The base model of the plasma torus used in this paper yielded a linear correlation coefficient of 0.540 between aurora brightness on Io and the plasma density at the position of the moon. After several additions to the model, the linear correlation improved and the best correlation coefficient obtained with the whole data set was 0.642. However, it cannot be ruled out that this improvement may just be a consequence of manipulating the plasma model to make it agree with the aurora brightness data and not an actual improvement of the model of the plasma torus. On the other hand, all changes made to the plasma model except the tuning of the parameters were motivated by physical arguments in accordance with previously established theory. The tuned parameters should be

seen as an upper bound for how closely our models can relate plasma density to brightness data and the models using parameters sourced from other studies are more trustworthy when considering accuracy for density measurements. Regardless, a Pearson coefficient of 0.540 for the base model is already a strong indication of the existence of a linear correlation in the data.

An interesting area for further study is the fact that the yearly divisions seem to indicate a linear dependence when analysed alone but that some time dependence interferes when analysing the data set as a whole. It is also notable that measurements on the dusk side of Jupiter seem to either be better modeled or follow the linear dependency significantly better.

APPENDIX A CONSTANTS IN EQUATIONS

APPENDIX B DERIVING THE DISTANCE ALONG MAGNETIC FIELD LINES

APPENDIX C OBSERVATIONAL DATA

ACKNOWLEDGMENT

The authors would like to thank the supervisor Lorenz Roth for the help throughout the project. Without his guidance we would not have understood the subject well enough to complete the project and if not for his insights we would have been left with multiple bugs and oddities that would have gone unnoticed.

REFERENCES

- [1] N. M. Schneider and F. Bagenal, "Io's neutral clouds, plasma torus, magnetospheric interaction," in *Io After Galileo: A New View of Jupiter's Volcanic Moon*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 265–286.
- [2] R. M. Lopes and D. A. Williams, "Chapter 43 - volcanism on io," in *The Encyclopedia of Volcanoes (Second Edition)*, second edition ed., H. Sigurdsson, Ed. Amsterdam: Academic Press, 2015, pp. 747–762. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780123859389000432>
- [3] F. Bagenal and V. Dols, "The space environment of io and europa," *Journal of Geophysical Research: Space Physics*, vol. 125, no. 5, p. e2019JA027485, 2020. [Online]. Available: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019JA027485>
- [4] W. H. Smyth, C. A. Peterson, and M. L. Marconi, "A consistent understanding of the ribbon structure for the io plasma torus at the voyager 1, 1991 ground-based, and galileo j0 epochs," *Journal of Geophysical Research: Space Physics*, vol. 116, no. A7, 2011. [Online]. Available: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2010JA016094>
- [5] P. Phipps and F. Bagenal, "Centrifugal equator in jupiter's plasma sheet," *Journal of Geophysical Research: Space Physics*, vol. 126, no. 1, p. e2020JA028713, Jan. 2021.
- [6] P. H. Phipps and P. Withers, "Radio occultations of the io plasma torus by juno are feasible," *Journal of Geophysical Research: Space Physics*, vol. 122, no. 2, pp. 1731–1750, Feb. 2017.
- [7] N. M. Schneider and J. T. Trauger, "The Structure of the Io Torus," vol. 450, p. 450, Sep. 1995.
- [8] NASA. (2022, Apr) The spice toolkit. [Online]. Available: <https://naif.jpl.nasa.gov/naif/toolkit.html>
- [9] F. Bagenal and R. J. Wilson, "Jupiter system iii (s3lh, s3rh)," in *Jupiter Coordinate Systems*, 12nd ed. Denver: University of Colorado, 2016, p. 4.
- [10] Kent State University Libraries. (2022, May) Spss tutorials: Pearson correlation. [Online]. Available: <https://libguides.library.kent.edu/SPSS/PearsonCorr>

Modelling of the Bow Shock and Magnetopause of Jupiter Using *In-situ* Juno Data

Lukas Grigelionis and Emanuel Reuthe Löfgren

Abstract—When the solar wind encounters a planet's magnetic field, they interact and the different plasma and magnetic field behaviours divides the magnetosphere into different regions. Two important region boundaries to the outer magnetosphere, called the magnetosheath, are the bow shock and magnetopause. A good deal of knowledge about the planet's magnetic field can be obtained by studying these boundaries. Moreover, the strength of Jupiter's magnetic field makes its magnetosheath boundaries an interesting case study. The aim of this study was to compile data covering the crossings of the Jovian bow shock and magnetopause from NASA's Juno probe and use this data to investigate their shape and location. In doing so, we hoped to be able to assess the validity of previous models and the stability of Jupiter's magnetic field over time. Both a parabolic curve model and a location distribution function were created as part of this objective. The distribution of boundary crossings prevented fine details in the shape and location of the bow shock and magnetopause from being determined. By analysing the density of occurring boundary crossings it was found that the bow shock and magnetopause are generally positioned closer to Jupiter than determined by previous studies.

Sammanfattning—När solvinden färdas nära en planet växelverkar de. Detta ger upphov till magnetosfären som består av olika områden med varierande plasmabeteenden. Två viktiga gränser till magnetosfärens yttre del, kallad magnetosheath, är bogchocken och magnetopausen. De är intressanta då man vid undersökning kan lära sig mycket om planets magnetfält överlag. Dessutom utgör gränserna av Jupiters magnetosheath en intressant fallstudie på grund av planetens starka magnetfält. Målet med denna studie var att sammanställa data över korsningar av Jupiters bogchock och magnetopause av NASA:s rymdsond Juno för att undersöka deras form och position. Med detta hoppades vi på att kunna bedöma validiteten hos tidigare modeller och bedöma stabiliteten av Jupiters magnetfält över tid. Både en parabolisk modellkurva och en fördelningsfunktion över gränsernas position skapades som en del av detta mål. Fördelningen av gränskorsningar förhindrade bestämmandet av mindre detaljer av form och position hos bogchocken och magnetopausen. Genom att analysera tätheten av förekommande korsningar upptäcktes det att bogchocken och magnetopausen befinner sig i allmänhet närmare Jupiter än vad som bestämts i tidigare studier.

Index Terms—Space, Jupiter, Bow Shock, Magnetopause, Juno, Plasma, Modelling.

Supervisors: Tomas Karlsson

TRITA number: TRITA-EECS-EX-2022:155

I. INTRODUCTION

A. Physics of the bow shock and magnetopause

At the sun's outermost layer, the corona, the star's gravity can no longer contain its rapidly moving constituent particles.

These particles then stream away from the star through space as solar wind. Due to the high temperatures, the corona is fully ionized and contains a mix of protons and electrons in the form of electrically conducting plasma according to [1]. As it travels through space, the solar wind eventually encounters the magnetic fields of various planets in the solar system. When this happens, the plasma and the magnetic fields interact with each other which gives rise to magnetospheres. The magnetosphere is separated into several regions in which the plasma-magnetic field interactions are different in their nature. The most interesting region in the context of this study is the magnetosheath, the outermost part of the magnetosphere as shown in Figure 1. The magnetosheath is bounded by the bow shock (BS) at its farthest points from the planet and the magnetopause (MP) at its closest points to the planet. Fälthammar [2] states that, at the BS, the plasma's velocity changes from supersonic to subsonic. Cairns [3] elaborates on what exactly happens to the plasma at the BS, stating that the solar wind flow is compressed, slowed, heated and deflected. These properties are characteristic of any BS, but the ones that occur around planets differ in one essential way.

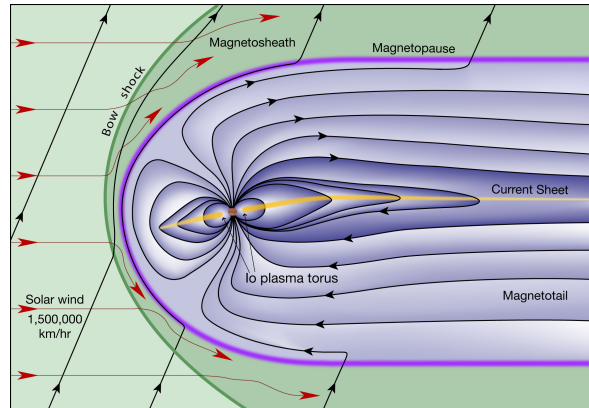


Fig. 1. An image of the magnetosphere of Jupiter with various regions marked, created by Bagenal and Bartlett [4]

A BS, such as that which can be observed in front of airplanes or bullets traveling at supersonic speeds, occurs due to collisions between molecules. Moreover, the thickness of such a BS is related to the average length of the free path of molecules between collisions. For the particles of the solar wind, that mean free path is usually on the scale of 10^{12} m, which implies almost total absence of collisions. Therefore, a traditional BS would be impossible. However, thanks to wave-particle interactions in the plasma, it is possible for a "collisionless" BS to occur. While the density of plasma is low enough that normal collisions don't occur, the electrical

charge of its constituent particles mean they can interact with electromagnetic fields. In particular, charged particles in plasma can interact with other charged particles over longer distances. Interactions between particles within plasma gives rise to electromagnetic waves in the plasma which in turn interact with other particles. The overall effect is that the plasma behaves more like an ordinary gas which is what makes a planetary BS possible according to Gedalin et al [5].

The magnetopause (MP), unlike the BS, is not a shock. However, it is nevertheless an important boundary and thus useful to map. The MP can be considered a discontinuity in plasma of the magnetosphere since the plasma density and pressure as well as the vector magnetic field vary across it [6]. Depending on the context, it can either be considered a rotational discontinuity, where the plasma flow and magnetic field retain their magnitude, but change in density or a tangential discontinuity, which means no particles flow over it.

B. Previous work

When it comes to modelling the MP and the BS, comprehensive models of these phenomena have been made for several planets such as Earth by Formisano [7], Mercury by Winslow et al [8] and Jupiter by Huddleston et al [9]. Jupiter has the strongest magnetic field in the solar system and the largest magnetosphere as well, an annotated image of which is shown in Figure 1. However, very little is known about the stability of Jupiter's magnetic field over time. As such, characterizing it would be an important step to understanding just how much different circumstances impact the nature of the magnetosphere. Modelling the MP and BS are a part of this process.

Although eight spacecraft have been launched on missions to Jupiter by NASA to date, the most recent one before Juno was launched in 1997 [10]. The latest two attempts at modelling the BS and MP were made in 1998 and 2002 by Huddleston, Russel and Kivelson et al [9] as well as Joy, Kivelson and Walker et al [11]. Both of these studies relied on data from Pioneer 10 and 11, Voyager 1 and 2 as well as Galileo and Ulysses with data from Cassini not being public at the time the models were made. The former model compiled data from six different missions and adapted them to a least squares fit. In contrast, the latter used Ogino-Walker magnetohydrodynamic simulations to create probabilistic models using much of the same data that the former model did. The most important conclusions of this study was that the phenomena are affected by pressure from the solar wind. Moreover, it is likely that the MP favors a bimodal distribution with two most likely positions: An outermost and an innermost. However, it has a range of locations based on the solar wind pressure despite this. The study of Joy et al [11] also notes that there is not enough evidence of the BS following such a distribution.

Since then, the Juno spacecraft was launched on a mission to Jupiter. The spacecraft arrived at the planet in 2016 and

has since been gathering various forms of data, including the electric and magnetic fields. This data, which is to be used in this study, has already helped characterize the magnetosphere in greater detail by Connerney et al [12]. Moreover, the boundaries (BS and MP) have also recently been studied to some degree by Hospodarsky et al [13] using Juno data, although not as meticulously as in the cases of Huddleston et al [9] and Joy et al [11]. The stability of Jupiter's magnetic field over time is also unknown and of great interest to the scientific community due to its extreme nature. As such, the goal of this study is to use data from Juno to build a model of the aforementioned boundaries. The data has been acquired *in situ* as it consists of measurements made by the probe while in orbit around Jupiter. The model is then to be compared to previous ones in order to assess the validity of previous models and analyze any potential changes in Jupiter's magnetic field over time. This in turn could serve as groundwork for future studies in this field of research.

TABLE I
ABBREVIATIONS

Meaning	Abbreviation
Bow Shock	BS
Magnetopause	MP
Root mean square	RMS
Crossing density	CD
Power spectral density	PSD
Planetary data system	PDS

II. METHODOLOGY

All figures and parameter values discussed are produced within the scope of the thesis work if not stated otherwise.

A. Compilation of data

Data of BS and MP encounters in the form of time, date and location relative to Jupiter at each crossing was gathered by analyzing Juno data from 2016-06-24 to 2021-06-09 using MATLAB. The data consisted of electric (E) and magnetic (B) measurements respectively from Juno's magnetic field instrumentation and Waves instrument. The former consists of a pair of independent sensor suites consisting of tri-axial fluxgate magnetometers with two collocated imaging sensors [14] while the latter consists of an electric dipole antenna and a magnetic search coil [15]. The data from these instruments was acquired via the Planetary Data System (PDS) [16]. The MATLAB tools used were sourced from the Swedish Institute of Space Physics at the University of Uppsala. Additional code based on these tools was also developed over the course of the study. Using these MATLAB tools data from Juno, stored as .sts and .csv files, can be visualized as graphs which can in turn be used to determine when crossings occur. Crossings usually coincide with drastic changes in the electric (E) and magnetic (B) fields, but changes in the former are usually more noticeable. Due to the E-field data taking up several gigabytes of memory and large parts of it being uninteresting within the scope of this project, possible crossings are found by first identifying "dates of interest" in the B-field data.

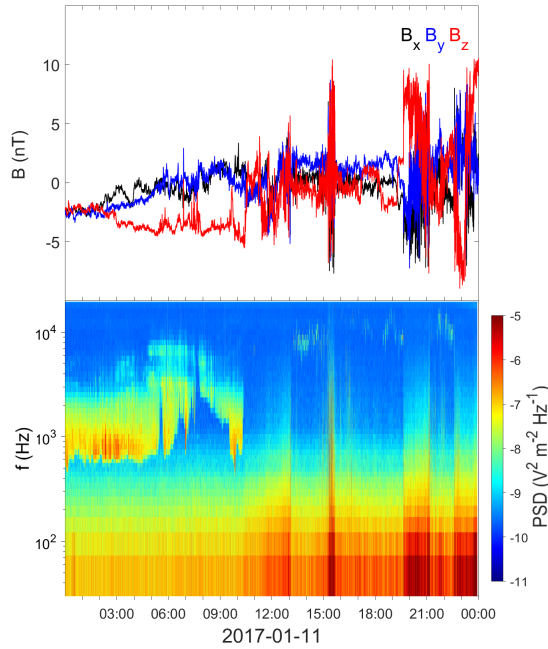


Fig. 2. Graph of magnetic field along sun-state coordinates and electric field spectrogram on 2017-01-11 with power spectral density following logarithmic scale on the right.

Jupiter's relatively fast rotation rate of 9 hours and 55 minutes combined with the planets magnetic poles being offset in respect to the axis of rotation creates periodic oscillations in the B-field when Juno is located inside the magnetosphere. This makes it easy to find dates of interest because these oscillations are absent when Juno is located within the magnetosheath and solar wind. The qualification for a date of interest is that the B-field data for that date has to contain clear changes in amplitude and frequency. What is meant by this is that one has to be able to observe distinct variations in the amplitude of the turbulence during a particular day for it to be considered a date of interest. In Figure 2, the B-field exhibits this exact behavior and thus 2017-01-11 can be considered a date of interest. Once a date of interest was determined, the E-field data for that date was downloaded and visually analyzed to confirm the existence of possible crossings. If any crossings of Jupiter's BS or MP were confirmed to occur within the analyzed time-frame, the time and location of the crossing(s) was documented.

B. Identification of crossings

The behavior of EM-fields are also measurably different within the various regions of the magnetosphere and the solar wind. In the inner magnetosphere, the power spectral density (PSD) is higher in the 0.5 to 9 kHz range compared to the magnetosheath and solar wind. Moreover, the PSD in the inner magnetosphere is also lower in the range 100 to 400 Hz compared to the magnetosheath and the solar wind. It's also worth noting that the PSD in this frequency range is higher in the magnetosheath than in the solar wind. Finally, the solar wind is also distinguished by a higher PSD around 10 kHz. This is due to Langmuir waves which are a type of plasma waves that are usually observed in the solar wind in the vicinity of a planetary BS according to Kellogg et al [17].

Documentation of the crossings was done in several steps. First, the points in time at which magnetic flux density and power spectrum density for the electric field change drastically in a way that indicated that Juno has crossed the BS or MP of Jupiter were identified. This is shown in Figure 2. Characteristic changes in field and frequency indicative of BS crossings occur at roughly 13:00, 15:15, 15:40, 19:40, 21:10 and 22:40 UTC, while corresponding changes indicative of a MP crossing occurs roughly at 10:30 UTC. Then, the orbit of the Juno spacecraft for the date of interest is plotted in MATLAB and the x , y and z coordinates at the time of the crossing are recorded. In this study, the sun-state coordinate system as described by Connerney [14] was used. More specifically, it is a three-dimensional Cartesian Jupiter-centered coordinate system which rotates with the planet along its orbit around the sun. The x -axis is always pointing towards the sun, the y -axis points in the opposite direction of orbital motion around the sun and the z -axis points to the celestial north. The units along these axes are normalized to the Jovian radius at the planet's equator $R_J = 71\,492\text{ km}$ [18]. A depiction of this coordinate system is seen in Figure 3.

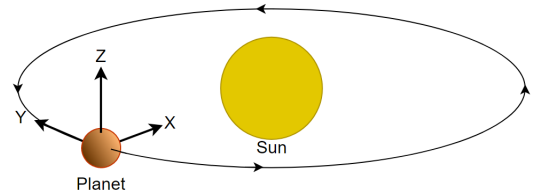


Fig. 3. The sun-state coordinate system

The most important orbital parameter for this study was the apoapsis, since a greater number of crossings will result in a more exact model. Since Juno's orbit around Jupiter was changing, parameters such as the eccentricity had little effect on how many crossings would occur. Once compiled, the data points were plotted so that they were superimposed on the orbits of the Juno spacecraft. The BS crossings can be seen in Figure 4 whereas the MP crossings can be seen in Figure 5. This was done to gain an understanding of where the crossings occur to determine if the model curves were reasonably accurate.

C. Modelling

Due to Juno's orbit all MP and BS crossings occurred on the morning side of Jupiter and therefor no fine details in the shape of these phenomena can be determined. As such, it was assumed that the BS and MP have a cylindrical symmetry around the x -axis which greatly simplified the model. However, due to the previously mentioned flatness of the Jovian magnetosphere, this model is only applicable to the region in space with close proximity to the plane of the Jovian equator. This is consistent with what was stated by Huddleston et al [9], that being that the bow shock and magnetopause can be assumed to be symmetrical over the equator. The mathematical expression that describes the model used in this study was described by Shue et al [19] and previously used

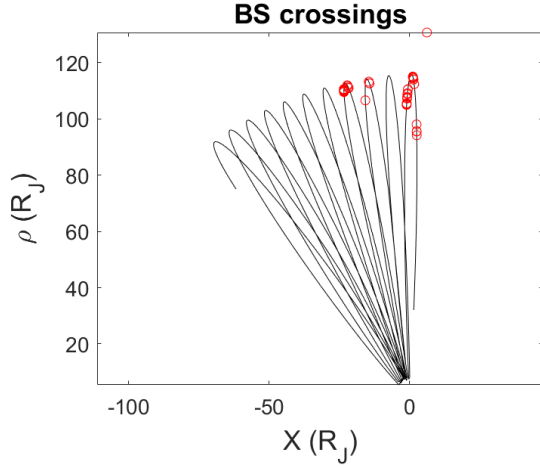


Fig. 4. First 11 orbits of Juno around Jupiter with BS crossings marked.

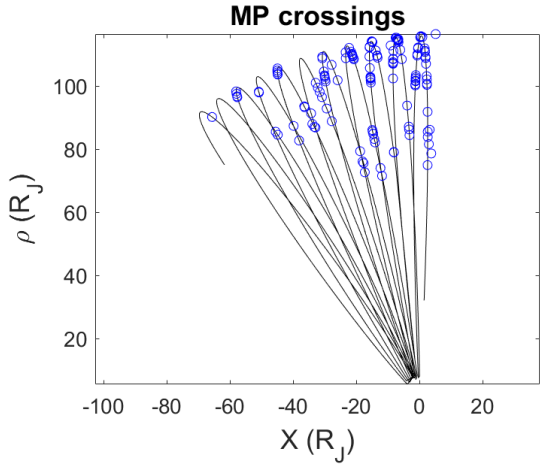
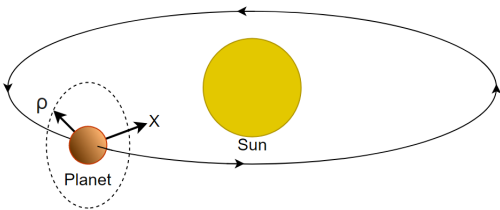


Fig. 5. First 11 orbits of Juno around Jupiter with MP crossings marked.

by Winston et al [8]. It is given by Equation 1, with R_{ss} in this context being given in units of the radius of Jupiter.

$$R(\theta) = R_{ss} \left(\frac{2}{1 + \cos(\theta)} \right)^\alpha \quad (1)$$

This model describes the distance R between Jupiter and the modeled shock phenomenon for a given angle θ between 0 and π from the x -axis. The collected data was mapped to the $x\rho$ -plane with $\rho = \sqrt{y^2 + z^2}$ being the radial coordinate in a cartesian to cylindrical transformation. A visualization of this cylindrical coordinate system is shown in Figure 6.

Fig. 6. Sun-state coordinates translated to cylindrical coordinates through $\rho = \sqrt{y^2 + z^2}$

The shape and location of the modelled boundaries could be determined by finding the values of the parameters R_{ss} and

α corresponding to the curve best fitted to the crossings. This was done by finding the minimal root mean square (RMS) value for the distance between the crossings and the boundary model curve. Specifically the RMS was calculated over a space spanning the parameters α and R_{ss} using equation 2 where P_n is the n :th out of N crossings in the chosen set of data and where Q_n is the nearest point to P_n on the model curve. By adjusting the values of R_{ss} and α , the resulting model curve would be different which in turn would affect the RMS value. The parameters whose model curve produced the lowest RMS value would best fit the data points and as such make for the most accurate model.

$$\text{RMS} = \sqrt{\frac{1}{N} \sum_{n=1}^N |P_n - Q_n|^2} \quad (2)$$

D. Modelling of phenomena location distribution

The solar wind can vary in speed and density which in turn causes the locations of the BS and MP with respect to Jupiter to vary. The expected location of these boundaries at any time will therefore follow a distribution which could be determined by analysing the distribution of phenomena crossings. In previous studies, changes in the distance between the MP, BS and Jupiter indicate the value of R_{ss} changes much more relative to that of α . This fact in combination with the relatively small distribution of crossings makes it suitable to approximate α as a constant which simplifies the calculation of the probability distribution of Juno crossing each boundary at any given range of R_{ss} . As is evident from Figure 2 several crossings could occur within a short period of time. This was likely due to the location of the phenomena oscillating as a result of variations in the solar wind. As it was thought that this might affect the CD calculations negatively, the data points were grouped in six hours to ameliorate this.

The location probability distributions of the boundaries were fitted to the density of crossings over Juno's orbit around Jupiter. This crossing density (CD) was evaluated by dividing the number of crossings in separate regions of the orbit by the area of the corresponding region. The calculation of the area of each region is further presented in Appendix B. The shape of the regions follow equation 1 with varying values of R_{ss} and α . However as stated previously, an approximation was used where the value of α is set to a constant which is presented in table #. Different values were chosen for the CD calculation of the MP and BS. Subsequently, the distribution function for the MP and BS locations were then fitted to the CD of the respective phenomenon. Of note is that this approach is novel, as no previous studies in this field have done anything similar.

III. RESULTS

The complete set of data available at the time of analysis where NASA's Juno probe approaches and orbits Jupiter span between 2016-01-05 and 2021-06-09. In total, respectively 69 and 155 crossings of the BS and MP were found. All crossings of the MP were found between 2016-06-25 and 2018-01-06,

and all crossings of the BS were found between 2016-06-24 and 2017-01-12. This corresponds to only the first 11 out of 34 available orbits containing crossings. A comprehensive list of all crossings of both types documented as part of this study can be found in Appendix A.

A. Model of inner bounds

As can be noticed in Figures 4 and 5 all crossings are confined within a specific region in the $x\rho$ -plane. The lower edge of this region forms a curve which can be regarded as the lowest distance in respect to Jupiter which the BS and MP respectively reach. Fitting an inner model to the crossings along this curve provides an approximate value for the minimal possible value for the parameters α and R_{ss} . Figure 7 shows the RMS value for different values of R_{ss} and α .

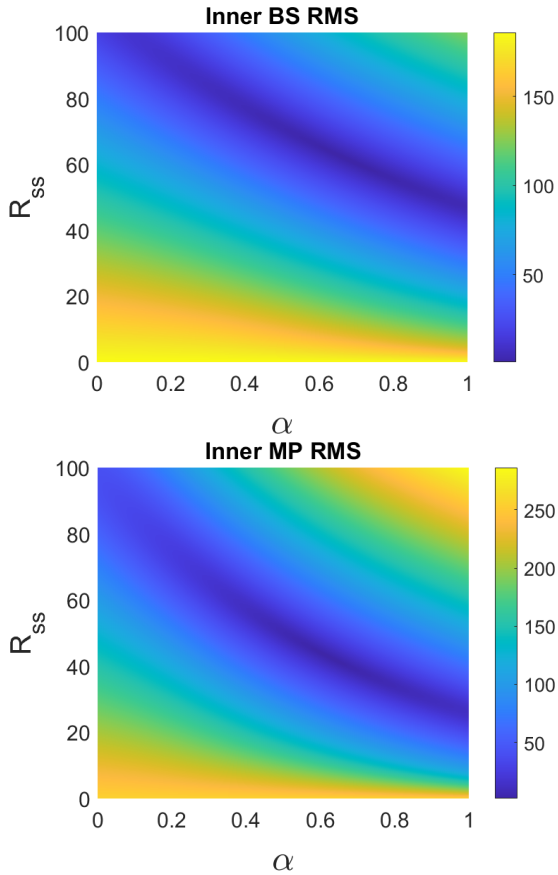


Fig. 7. RMS analysis corresponding to the innermost crossings of the BS and MP. Color bar on right corresponds to calculated RMS value in units of R_J

TABLE II
PARAMETER VALUES FOR MP AND BS MODEL CURVES

	R_{ss}	α
BS	$61.7117 R_J$	0.6724
MP	$42.9229 R_J$	0.6207

TABLE III
SHORTEST DISTANCE FROM JUPITER TO STUDIED BOUNDARIES

	Huddleston [9]	Joy (90th percentile curve) [11]	Junio
BS	$65 R_J$	$62.5 R_J$	$61.7117 R_J$
MP	$45 R_J$	$46.5 R_J$	$42.9229 R_J$

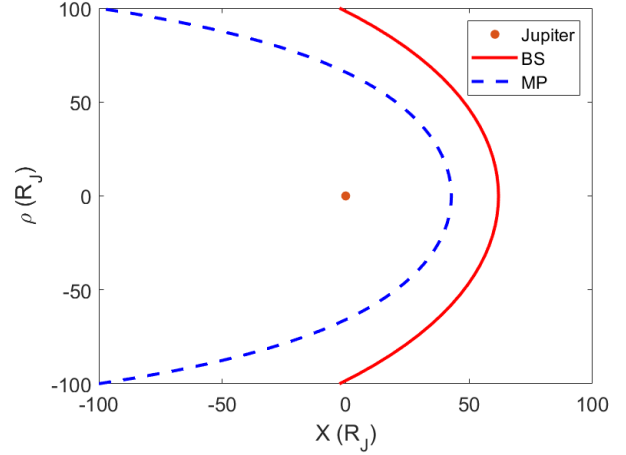


Fig. 8. Models corresponding to the best fit for innermost possible MP and BS curves

B. Model of shock location distribution

The CDs that resulted from the integral calculation can be seen in Figure 9. They were also fitted to normal and skew normal distributions with corresponding coefficients listed in Table IV. This is because the probability distribution functions of many phenomena can be approximated as normal distributions provided there are enough samples due to the central limit theorem.

TABLE IV
DISTRIBUTION COEFFICIENTS

	BS	BS (skew)	MP	MP (skew)
μ	73.60	80.93	71.17	77.18
σ	6.594	9.853	7.097	11.74
λ	—	-2.444	—	-5.105

IV. DISCUSSION

A. Comparison with previous work

Figure 8 shows a RMS optimized model based on the innermost crossings found as part of this study. In comparison to Joy et al [11] and Huddleston et al [9], the model is simplified and only two-dimensional due to assumptions of rotational symmetry around the x -axis. Therefore, the most suitable comparisons to make between older models and this are in the xy -plane. Specifically, the point closest on the boundaries to Jupiter (at $y = 0$ in previous models and $\rho = 0$ in the model of this study) is the one most suitable for direct comparisons. Visual examination yields the results shown in Table II which can be compared to the R_{ss} values in Table III. Based on this, it seems as if the model in this study indicates that the BS and MP may be closer to Jupiter than previously thought. However, this is uncertain for reasons discussed in Section IV-B. In this context, a probe will have encountered the phenomena with a 90 % likelihood once it passes the 90th percentile curve indicates the line.

In Figure 7, an overview of the RMS error of the data points for innermost boundary positions can be seen for different values of α and R_{ss} . Although there are certain parameters that gave an optimized curve relative to the data used in

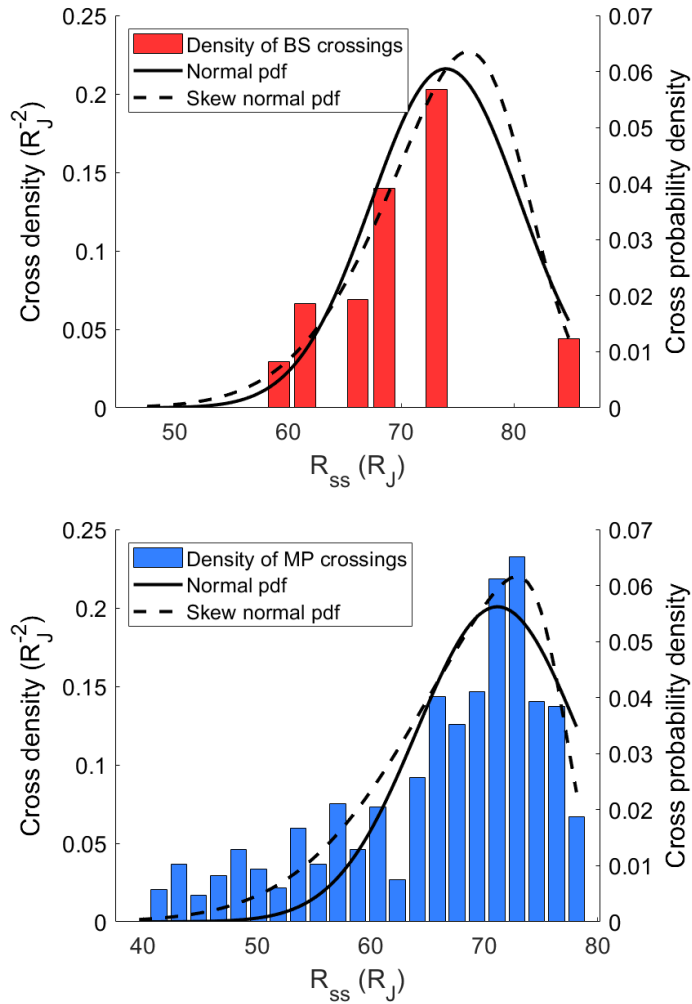


Fig. 9. Crossing density of BS and MP, alongside normal distribution and normal distribution fitted to crossings.

this study, that curve is only a best fit of all the points in this data set. Since the crossings only occurred on the morning side of Jupiter and are few in number, there is a possibility that the BS and MP are located at a different distance from Jupiter, but also possess different shapes. Even under the assumption of cylindrical symmetry, α and R_{ss} can vary to some degree while still producing well-fitting curves. As can be seen in Figure 7 the slightly dark region is thinner for the MP than for the BS. This suggests that the MP's location is more likely to be accurate in this study's model than the BS model. This is likely due to the fact that Juno crossed the MP more than twice as many times as it crossed the BS in the time interval examined in this study.

Concerning the CDs of the BS and MP, they do little in terms of explaining the results, but may partially validate the results of previous studies. Figure 9 shows that the data covering Juno's BS crossings is hard to fit accurately to a normal distribution. The MP crossing density is more similar to a normal distribution, but doesn't fit that description exactly. Neither of the CDs for either of these two boundaries seem to indicate a bimodal distribution as described by a

previous study. The skewed distribution that was made with the intent to include the outlying points of data better also gave questionable results. In addition to this, it's worth noting that the mean and variance of the CD for both boundaries are relatively high. It also contrasts Figure 8 which places the boundaries comparatively close to Jupiter. Moreover, it indicates that the phenomena are more likely to be found closer to Joy's 50th percentile curve. This may be a consequence having relatively little data to work with, in particular when it comes to fitting the data to a normal distribution. The possibility of approximating the distribution of the MP's and BS's locations as normal distributions with the central limit theorem was likely overly optimistic. However, it remains difficult to determine how valid these results are and by extension what this means for previous models and the stability of Jupiter's magnetic field.

B. Sources of error

There are many reasons which have negatively impacted the model's accuracy. First of all, the study had to be completed within a relatively short period of time which meant that the effects of the solar wind could not be accounted for. This was primarily because there was not enough time to understand how to extract relevant information from other types of data gathered by Juno. Therefore, the dynamic pressure from the solar wind could not be calculated which is essential to understand how it affects the location and shape of the MP or BS. In lieu of that, the CDs were made to see what the most probable locations were. However, as nothing like this calculation has been done in previous studies in this field, this approach may not have been entirely suitable. Moreover, by grouping crossings that happened within six hour intervals, the accuracy of the CD calculations was affected. This is because Juno moves slower the farther away from Jupiter it is. As a result, the method used to reduce the effect rapid changes in phenomena location can have on the CD calculations becomes less effective at longer distances from Jupiter. This may go some way to explain the high means in the CD of the MP and BS.

Second of all, the sample size of crossing was small and had a small spread which made the model less accurate. All the crossings occurred on the side of Jupiter that faces the sun and the number of crossings was small, particularly in the case of the BS. None of the encounters also occurred in the parts of the BS and MP that are the closest to the planet. The fact that no data from earlier missions could be included due to time limitations compounded this issue. For comparison, previous models both used data from several complete missions whereas this study was only working with data from a mission which, as of April 2022, is still ongoing. Last of all, Jupiter also orbits around the sun which rotates the BS and MP away from the orbit of Juno into the magnetotail. This fact combined with Juno not having a high enough apoapsis around Jupiter to reach parts of the MP and most of the BS further limits the number of crossings observed. Overall, a more accurate BS and MP model could be made if both data sets from previous

missions and a complete data set from the Juno missions could be used.

C. Future work

Given these facts, future researchers will have to address the aforementioned problems, the most important of which is the problem of sample size. In order to create as comprehensive and certain a model as possible, one needs to use as much data as possible. At the time of writing, Juno was still gathering data. Based on the probe's current orbit, Juno should emerge from the magnetotail on the evening side of Jupiter some time between May and June 2022. This may however be delayed due to the apoapsis of Juno being reduced over time. In general, future studies will have more data from Juno to work with and, if given time, could combine it with data from previous missions to Jupiter. This would provide as large a data set as possible could be used to make a more detailed model. Given that the missions from which data could be sourced have occurred over a number of decades, changes the location and shape of the MP and BS over time could also be taken into account by such a model. This in turn could help assess the stability of Jupiter's magnetic field in a more detailed way than simple comparisons between studies using data from different missions.

The second most important problem is the question of external factors affecting the results of this study. In particular, the particle data was formatted in such a way that it was not possible to completely analyze them within the time allotted to this study. As such the dynamic pressure of the solar wind affecting the MP and BS, accounted for by previous studies, could not be determined. As such, studies who have the benefit of more time could investigate this and perform the necessary calculations. This in turn could be used to create a mathematical function based that takes this into account as well as possible any possible asymmetries that were disregarded by this study. This could be done by looking at the problem in a 3D environment using Cartesian coordinates, which could also produce a model more suitable for direct comparisons with previous work. All in all, the location and shape of the MP and BS of Jupiter could be modelled in far greater detail by later work.

V. CONCLUSIONS

In summary, this study modelled the MP and BS of Jupiter using MATLAB and statistical data compiled from measurements of the Juno spacecraft. The purpose was to use the more sophisticated measuring equipment of the spacecraft to expand on previous work done using older technology. The model used for this purpose was a two-dimensional parabolic curve which was compared visually to parts of previous models. In addition to this, the CD of the BS and MP were both determined in order to analyze the likelihood of the phenomena being found at certain locations. Although the BS and MP models were somewhat similar to older ones, it is hard to make rigorous claims about the validity of the

various models. The most notable difference between this model and the older one is that the boundaries appear to be somewhat closer to Jupiter than in previous models. The CDs highlight the scarcity of data and also indicate that the location distribution of the phenomena don't fit normal distributions particularly well. In the case of the MP, this may indicate that its position follows a bimodal distribution, although it is not certain. As concerns the BS, there is too little data on it to draw any conclusions on the distribution of its position. Overall, these results means it is hard to make any conclusive statements about any changes in the shape and location of the BS and MP, the validity of previous models or the stability of Jupiter's magnetic field. Nevertheless, this study may serve as a stepping stone towards a better model in the future.

APPENDIX A

TABLE OF MAGNETOPAUSE AND BOW SHOCK CROSSINGS

APPENDIX B

CD AREA CALCULATION

ACKNOWLEDGMENT

The authors would like to thank Tomas Karlsson for his helpful and much appreciated support and supervision during the course of this bachelor thesis project.

OPEN RESEARCH STATEMENT

Data analysis was performed using the IRFU-Matlab analysis package available at <https://github.com/irfu/irfu-matlab>. Additional code utilizing this package as well as crossing statistics documented within the thesis work can be found at <https://github.com/DalecarliaAstro/KEX-K2-2022>.

REFERENCES

- [1] C.-G. Fälthammar, *Space physics*. Sweden, Stockholm: Royal Institute of Technology, 1992, pp. 122–124.
- [2] —, *Space physics*. Sweden, Stockholm: Royal Institute of Technology, 1992.
- [3] I. Cairns. (1999, Sep.) Basic physics of the bow shock. University of Sydney, Camperdown NSW 2006, Australia. [Online]. Available: <http://www.physics.usyd.edu.au/~cairns/teaching/lecture13/node2.html>
- [4] F. Bagenal and S. Bartlett. (2013) Magnetospheres of the outer planets group: Graphics. University of Colorado Boulder. [Online]. Available: <https://lasp.colorado.edu/home/mop/resources/graphics/>
- [5] M. Gedalin, C. T. Russell, and A. P. Dimmock, "Shock mach number estimates using incomplete measurements," *Journal of Geophysical Research: Space Physics*, vol. 126, no. 10, 2021, <https://doi.org/10.1029/2021JA029519>.
- [6] M. G. Kivelson and C. T. Russell, *Introduction to space physics*. Cambridge university press, 1995.
- [7] V. Formisano, "Orientation and shape of the earth's bow shock in three dimensions," *Planetary and Space Science*, vol. 27, no. 9, pp. 1151 – 1161, 1979.
- [8] R. M. Winslow, B. J. Anderson, C. L. Johnson, J. A. Slavin, H. Korth, M. E. Purucker, D. N. Baker, and S. C. Solomon, "Mercury's magnetopause and bow shock from messenger magnetometer observations," *Journal of Geophysical Research: Space Physics*, vol. 118, no. 5, pp. 2213 – 2227, 2013.
- [9] D. Huddleston, C. Russell, M. Kivelson, K. Khurana, and L. Bennett, "Location and shape of the jovian magnetopause and bow shock," *Journal of Geophysical Research: Planets*, vol. 103, no. E9, pp. 20075 – 20082, 1998.
- [10] D. R. Williams. (2019, Oct.) Jupiter. National Aeronautics and Space Administration. [Online]. Available: <https://nssdc.gsfc.nasa.gov/planetary/planets/jupiterpage.html>

- [11] S. Joy, M. Kivelson, R. Walker, K. Khurana, C. Russell, and T. Ogino, "Probabilistic models of the jovian magnetopause and bow shock locations," *Journal of Geophysical Research: Space Physics*, vol. 107, no. A10, pp. SMP-17-1 – SMP-17-17, 2002.
- [12] J. Connerney, S. Timmins, R. Oliverson, J. Espley, J. Joergensen, S. Kotsiaros, P. Joergensen, J. Merayo, M. Hecceg, J. Bloxham *et al.*, "A new model of jupiter's magnetic field at the completion of juno's prime mission," *Journal of Geophysical Research: Planets*, vol. 127, no. 2, 2022, <https://doi.org/10.1029/2021JE007055>.
- [13] G. Hospodarsky, W. Kurth, S. Bolton, F. Allegrini, G. Clark, J. Connerney, R. Ebert, D. Haggerty, S. Levin, D. McComas, C. Paranicas, A. Rymer, and P. Valek, "Jovian bow shock and magnetopause encounters by the juno spacecraft: Juno bow shock and magnetopause encounters," *Geophysical Research Letters*, vol. 44, pp. 4509 – 4511, May 2017.
- [14] J. Connerney, M. Benn, J. Bjarno, T. Denver, J. Espley, J. Jorgensen, P. Jorgensen, P. Lawton, A. Malinnikova, J. Merayo *et al.*, "The juno magnetic field investigation," *Space Science Reviews*, vol. 213, no. 1, pp. 39–138, 2017.
- [15] W. Kurth, G. Hospodarsky, D. Kirchner, B. Mokrzycki, T. Averkamp, W. Robison, C. Piker, M. Sampl, and P. Zarka, "The juno waves investigation," *Space Science Reviews*, vol. 213, no. 1, pp. 347–392, 2017.
- [16] National Aeronautics and Space Administration. (2022, Apr.) Juno data holdings. [Online]. Available: https://pds-ppi.igpp.ucla.edu/search/?sc=Juno&facet=SPACECRAFT_NAME
- [17] P. J. Kellogg, "Langmuir waves associated with collisionless shocks; a review," *Planetary and Space Science*, vol. 51, no. 11, pp. 681–691, 2003.
- [18] P. K. Seidelmann, B. A. Archinal, M. F. A'hearn, A. Conrad, G. Consolmagno, D. Hestroffer, J. Hilton, G. Krasinsky, G. Neumann, J. Oberst *et al.*, "Report of the iau/iag working group on cartographic coordinates and rotational elements: 2006," *Celestial Mechanics and Dynamical Astronomy*, vol. 98, no. 3, pp. 155–180, 2007.
- [19] J.-H. Shue, J. Chao, H. Fu, C. Russell, P. Song, K. Khurana, and H. Singer, "A new functional form to study the solar wind control of the magnetopause size and shape," *Journal of Geophysical Research: Space Physics*, vol. 102, no. A5, pp. 9497–9511, 1997.

Using Satellite Data to Calculate Entropy of Electrons at Collisionless Shocks

Alice Wallner and Sofie Berglund

Abstract—The solar wind is a supersonic flow of protons and electrons emitted in all directions from the sun. As the supersonic solar wind encounters Earth’s magnetic field, it creates the Earth’s bow shock, which increases the kinetic entropy of electrons passing through it. In this study, the aim is to analyze shock crossings of Earth’s bow shock in order to draw conclusions of which shock parameters that are important for kinetic entropy generation. Due to knowledge gained from an earlier study by M. Lindberg et al. [1], the shock crossings of interest in this study are quasi-perpendicular shocks with a low electron plasma beta. The data used is measured with the NASA MMS spacecraft and accessed through IRF Uppsala. As a result, a database with 13 shock crossings was created and the entropy change was related to, among other parameters, temperature and density change, shock angle, Alfvén Mach number, ion ram pressure and upstream magnetic field. We found that a high Alfvén Mach number related nearly proportionally to a large change in electron entropy for low electron plasma beta quasi-perpendicular collisionless shock crossing.

Sammanfattning—Solvinden består av protoner och elektroner som emitteras ut från solen i alla olika riktningar med enorma hastigheter. När dessa partiklar, med en hastighet som överstiger signalhastigheten, träffar Jordens magnetfält uppstår Jordens bågchock. Bågchocken ökar den kinetiska entropin hos elektroner som färdas genom den. För den här studien är målet att analysera chockkorsningar vid Jordens bågchock för att kunna dra slutsatser om vilka chockparametrar som är viktiga för generering av kinetisk entropi. Till följd av en tidigare studie av M. Lindberg et al. [1] är det endast kvasi-vinkelräta chockkorsningar med ett lågt plasma beta som denna studie avser. Den uppmätta datan erhålls från NASAs MMS satelliter och kan nås genom IRF Uppsala. Resultatet blev en databas med 13 chocker där entropiförändringen plottades mot bl. a. temperatur- och densitetsändring, chockvinkel, Alfvén Machtal, jontrycket och magnetfältet uppströms. Det upptäcktes då att ett högt Alfvén Mach-tal indikerade på en stor entropiökning hos elektroner vid kollisionslösa, kvasi-vinkelräta chockkorsningar med låga elektronplasmabeta.

Index Terms—entropy, electrons, plasma, Earth’s bow shock, magnetic field, collisionless shock, plasma beta, Alfvén Mach number, MMS, NASA, satellite, IRF

Supervisors: Martin Lindberg, Andris Vaivads

TRITA number: TRITA-EECS-EX-2022:156

I. INTRODUCTION

In a vacuum-like environment, phenomena does not always behave in the same way as they do in the atmosphere surrounding us on Earth. As a matter of fact, the vacuum of space is actually filled with a type of ionised “gas”, plasma. When plasma comes in contact with magnetic fields in space, the ionised particles are affected due to their electrical charge,

which causes significant change in the particles properties. This happens at Earth’s bow shock, where particles from the solar wind encounters the Earth’s magnetic field. The bow shock is known as a collisionless shock wave in which electrons passing through obtain a change in entropy larger than zero [2]. Entropy generation, a measure of disorder, is usually based on collisions between particles but, in collisionless shocks, the interactions between waves and particles, according to D. A. Tidman and N. A. Krall in [3], generate entropy - without such a well defined cause.

This type of entropy generation, has been studied by, among others, G. K. Parks et al. using the Cluster spacecraft [2]. In a more recent study, by M. Lindberg et al. [1] using the MMS spacecraft, it was found that a low upstream electron plasma beta indicates a large change in entropy for electrons. The electron plasma beta is the ratio between the particle pressure and the magnetic pressure. By gaining this knowledge, we do not just gain knowledge of entropy, but of irreversible heating processes and the energy distribution throughout them.

As shown in Fig. 1, there are two different regions of shock crossings around Earth’s bow shock, known as quasi-perpendicular and quasi-parallel shocks. The angle between the shock normal and the upstream magnetic field determines which region it should be classified as. Angles between 45° and 90° are classified as quasi-perpendicular shock crossings and will be studied in this paper. Smaller angles than 45° are classified as quasi-parallel shocks.

In this study, the relation between different shock parameters and the change in electron kinetic entropy is investigated for electrons crossing Earth’s bow shock with a quasi-perpendicular angle. This is a continuation study of a paper by M. Lindberg et al. [1]. Therefore, the investigation in this study will only focus on shocks with an upstream $\beta_e < 1$, in order to see correlation for other parameters important for entropy generation.

II. THEORY

A. Shocks, with and without collision

A shock wave can appear both within the Earth’s atmosphere and outside of it. In Earth’s atmosphere this occur when an object travels faster than the local speed of sound, known as breaking the sound barrier. The phenomena gives off a loud sonic boom. The physical process can be explained by the media in front of the object being violently compressed, making the particles collide with each other. In order to classify the intensity of such a shock wave, the Mach number

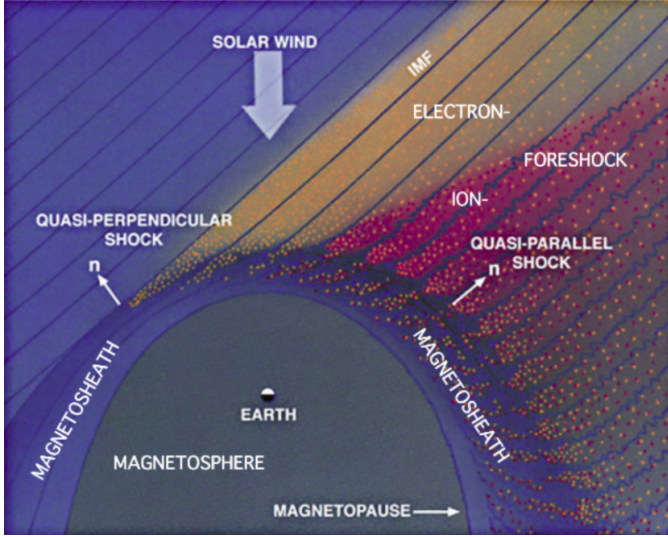


Fig. 1. An illustration of the bow shock and solar wind and where the quasi-perpendicular and quasi-parallel shock regions are. Picture from C. Kennel et al. in [5].

was defined, by Ernst Mach. The Mach number indicates whether a shock wave occurs or not. It is defined as;

$$M = \frac{u}{c_s}, \quad (1)$$

where u is the relative velocity between the object and upstream flow and c_s is the local speed of signal transmission, in this case, the speed of sound. Collisionless shock waves display similar behavior of violent compression, but without collisions between the particles in the media. These type of shocks are typically found outside of Earth's atmosphere, in space, which is often considered a perfect vacuum. Although this is not the case since space is filled with charged particles like ions and electrons. Charged particles forming an ionized gas known as plasma, the fourth state of matter. Space plasma is a dilute type of matter with such low density that the shock waves that occur when an object's speed is greater than the sonic speed does not mainly depend on collisions between particles. In such shocks, the energy and temperature changes are similar as shocks with collisions, but is here also dependent on the magnetic field properties.

The bow shock studied in this paper is the Earth's bow shock. This shock appears between Earth's magnetic field and the solar wind. An illustration of this can be seen in Fig. 1. The bow shock is studied using data from the Magnetospheric multiscale (MMS) spacecraft [4] orbiting Earth in equatorial orbit. The NASA MMS mission consists of four satellites in a tetrahedral formation, traveling in and out of Earth's magnetosheath while observing magnetic reconnection and measuring electromagnetic fields and the properties of ions and electrons found in the plasma environments of the Earth's magnetosphere. The magnetosheath is located, as can be seen in Fig. 1, between the magnetopause and the bow shock.

B. Shock parameters

This section lists the shock parameters relevant for this thesis. These include; the Alfvén Mach number M_A , a ratio

between normal component to the upstream flow velocity $V_{n,u}$ and the Alfvén speed V_A :

$$M_A = \frac{V_{n,u}}{V_A}, \quad (2)$$

where the Alfvén speed is calculated as follows

$$V_A = \frac{B}{\sqrt{\mu_0 n_i m_i}}, \quad (3)$$

where m_i and n_i is the ion mass and ion density, respectively.

As mentioned earlier, another important property when analyzing a shock crossing is the upstream electron plasma beta. It is a unitless quantity calculated by

$$\beta_e = \frac{2n_e k_B T_e \mu_0}{B^2}, \quad (4)$$

where k_B represents Boltzmann's constant, μ_0 is the permeability of vacuum, n_e is the number density of electrons, T_e is the electron temperature - a weighted sum of parallel and perpendicular temperature - and B the magnetic field. Essentially, the β_e is the ratio of the particle pressure and the magnetic pressure in the plasma.

θ_{Bn} , the angle between the shock normal and the upstream magnetic field,

$$\theta_{Bn} = \arccos \left(\frac{\mathbf{B}_u \cdot \hat{\mathbf{n}}}{|\mathbf{B}_u|} \right), \quad (5)$$

referred to as the shock angle.

The whistler Mach number M_{wh} as defined in A. Lalti et al. [6] and M. Oka et al. [7] as

$$M_{wh} = \frac{1}{2} \sqrt{\frac{m_i}{m_e}} |\cos \theta_{Bn}|. \quad (6)$$

This property will only be used in relation to the Alfvén Mach number, where $\frac{M_A}{M_{wh}} < 1$ indicates the possibility for standing Whistler waves in front of the shock.

The ion ram pressure

$$P_{i,ram} = \frac{1}{2} n_{i,u} m_i V_{n,u}^2, \quad (7)$$

where $n_{i,u}$ is the upstream ion density, m_i the ion mass and $V_{n,u}$ the upstream velocity in the shock normal direction.

The electron temperature anisotropy in the solar wind, defined as

$$A_{e,sw} = \frac{T_{perp}}{T_{par}} - 1, \quad (8)$$

where T_{perp} is the perpendicular electron temperature and T_{par} is the parallel.

Lastly, the change in electron temperature ΔT_e and number density Δn_e . All of these parameters will be calculated and analyzed in relation to the change in electron entropy, ΔS_e .

C. Categorizing the structures of shocks

When analyzing a quasi-perpendicular shock crossing, different parts of the shock crossing can be identified. The regions pictured in Fig. 2, bottom panel, include; upstream, shock foot, shock ramp, overshoot, undershoot and downstream. The upstream region represents the solar wind and the downstream

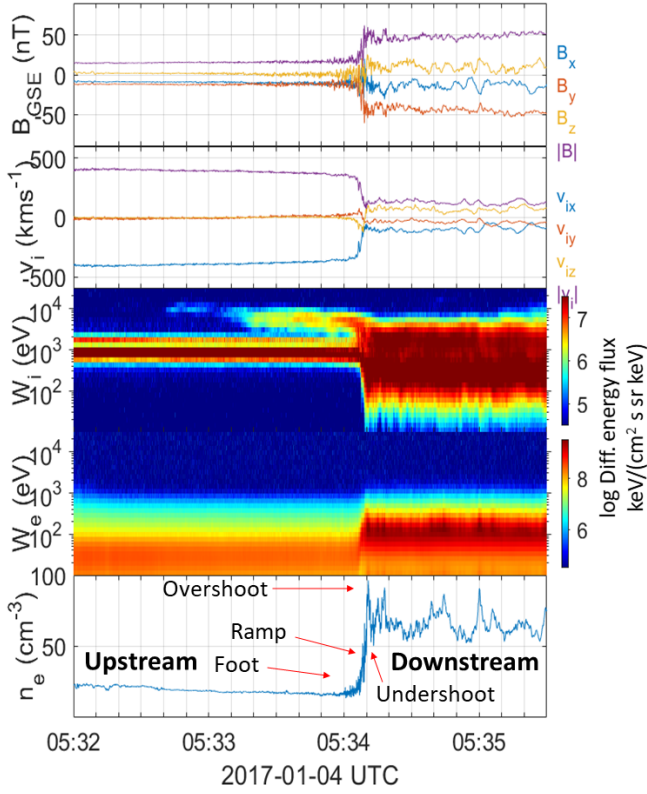


Fig. 2. Example of a quasi-perpendicular shock crossing. Panels from the top show; magnetic field, ion velocity, omnidirectional ion energy flux, omnidirectional electron energy flux and electron density. In the bottom panel, the different parts of the shock are marked.

region the magnetosheath. Foot, ramp, overshoot and undershoot are all part of the transition layer between up- and downstream. The ramp, the steep rise or fall in all of the panels, is the common denominator for all quasi-perpendicular shock crossings and is used when identifying the time of the shock. Moreover, the transition layer is not used when calculating the shock parameters since these are only based on a mean value of the upstream respectively downstream region.

This shock in particular, occurring on April 2017 has a shock angle of 57° and an electron plasma beta of 0.9. Its Alfvén Mach number is 8.3. One can easily see the impact the bow shock has on the properties of the particles. For instance in the ion velocity, where the ions initially are traveling with a speed of nearly 400 km/s and after encountering the bow shocks decelerates down to approximately 100 km/s. What can be interpreted as a sort of tail (around 10 keV) in the omnidirectional ion energy flux, is actually reflected ions from the bow shock.

D. Entropy

An important property when studying shocks is the concept of entropy. Entropy is a measure of disorder in thermodynamics and can be used to determine the direction, or the chronology, of events. In a closed system, the change in entropy will always be positive according to the second law of

thermodynamics. The same goes for any irreversible heating process, discussed in [8].

In order to find the entropy change for a shock crossing, the upstream region is analyzed and compared to the downstream region. This is done by calculating the entropy by particle according to:

$$S = - \frac{k_B \int f \ln f d^3\mathbf{v}}{\int f d^3\mathbf{v}}, \quad (9)$$

defined by H. Liang et al. [9], where S represents the kinetic entropy per electron and f is the distribution function for electrons from the MMS spacecraft data.

III. METHOD

A. Data and software

In order to further investigate how the entropy change is related to the different plasma properties, satellite data of the magnetic field, the electric field and the distribution function is needed. The instruments on the MMS spacecraft used to obtain this data is: the fluxgate magnetometer (FGM) [10], the fast plasma investigation (FPI) [11], the spin plane double probe (SDP) [12] and the axial double probe (ADP) [13]. The data from the MMS spacecraft mission is accessed through the MMS SDC mirror at IRF, Swedish Institute of space physics, Uppsala. By using the database created by A. Lalti et al. [14], it is possible to sort through the shock data based on the time of the shock and other associated parameters.

Software used throughout this project to search and calculate is Matlab 2020b from [15]. The Secure Shell protocol is used to mount the IRF drive and in order to perform calculations and data analysis, the IRFU-Matlab analysis package [16] is used.

B. Shock search algorithm

A search algorithm is developed in order to find shock crossings that satisfies the following conditions: a time interval of 2016-06-01 until 2020-12-31, a quasi-perpendicular shock crossing ($\theta_{Bn} > 45^\circ$) and a low electron plasma beta ($\beta_e < 1$). The time interval is chosen in order for the correction methods, implemented by M. Lindberg et al. [1], to function later for calibration. Due to that the values change slightly when calibrating, it was decided to narrow down the search interval further to $\theta_{Bn} > 60^\circ$ and $\beta_e < 0.9$.

For time efficiency purposes regarding the download of data required to calculate β_e , the algorithm is split into two parts. The first part searches through the A. Lalti's database [14] for shock crossings in the specified time interval and shock angle whereas the second part calculates β_e only for the shock crossings found in the previous part. As a final step, the obtained shock crossings are checked manually to ensure the conditions are met.

C. Corrections of the electron distribution function

Due to spacecraft interference with the plasma, corrections must be made to the MMS data. Moreover, an extrapolation of the distribution function down to zero energy must also be made since the spacecraft does not measure in this range.

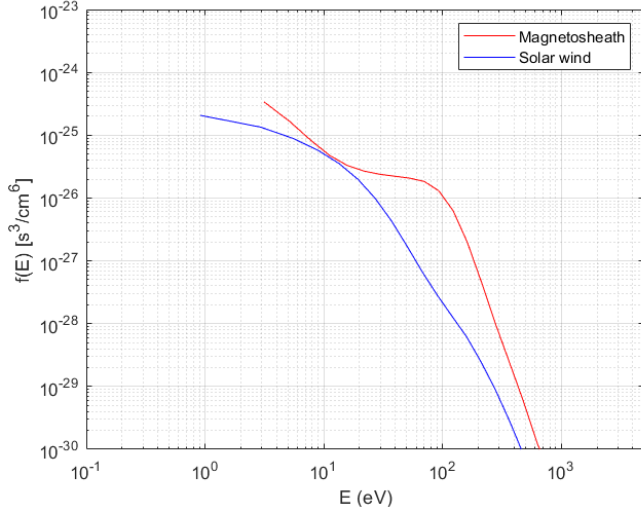


Fig. 3. Measured distribution functions in the solar wind (blue) and in the magnetosheath (red).

All four steps described in this section are according to M. Lindberg et al. [1].

Because of sun radiation and particle bombardment the spacecraft will emit photoelectrons and secondary electrons, thus inducing a potential Φ , as stated by D. J. Gershman et al. [17]. As a result, the electrons and ions in the plasma will be accelerated/decelerated and measured at a higher respectively lower energy E' , described as

$$E' = E - q\Phi, \quad (10)$$

where E is the actual energy of the particle, q is the charge of the particle and Φ is the spacecraft potential. By using the spacecraft potential measured by the electric field instruments on the spacecraft and equation (10), the shift in energies are corrected for.

The emitted electrons influence not only the spacecraft potential, but also contaminate the measured distribution function. The contamination is mostly affecting the lower energy channels on the MMS spacecraft, which can be seen as an upswing in Fig. 3 in the range $E < 20$ eV.

In order to correct the electron distribution function, a method described in M. Lindberg et al. [1], is used. The method is based on the conditions that the distribution function should resemble a Maxwellian distribution in the solar wind and a flat top distribution in the magnetosheath and by estimating the density of secondary electrons, the secondary electron contamination can be corrected for. An example of the distribution function after corrections is shown in Fig. 4.

When calculating the entropy, the whole range of the distribution function is needed to ensure a better estimate. As mentioned earlier, the MMS spacecraft does not measure the near-zero energy particles and therefore an extension must be made. The method, established by M. Lindberg et al. [1], is based on the assumption that the distribution of the electrons in the low energy range has little variation in phase space and therefore a linear extrapolation from the lowest value to zero energy is made which can also be seen in Fig. 4.

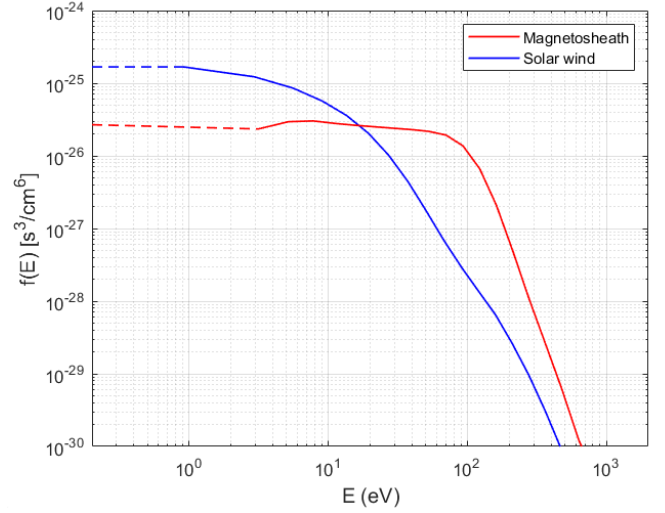


Fig. 4. Corrected distribution functions for the solar wind (blue) and the magnetosheath (red) with extrapolation to zero energy (dotted lines).

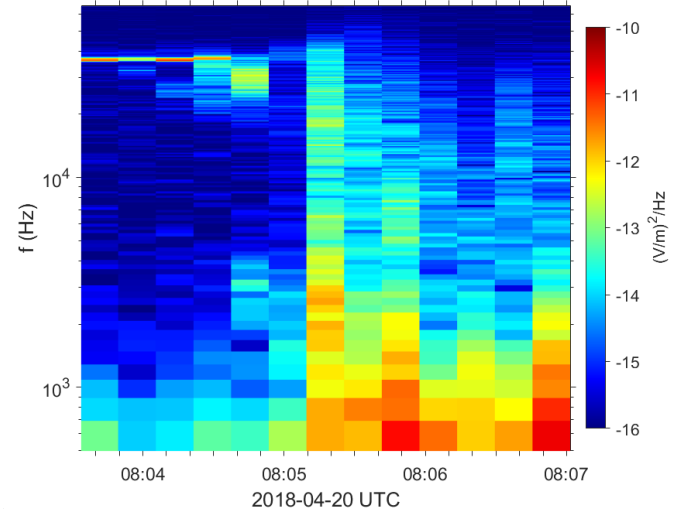


Fig. 5. Plasma line (around $f \sim 35$ kHz) representing the plasma frequency.

The final calibration of the distribution function is done by using the plasma frequency f_{plasma} to calculate the electron density in the solar wind $n_{e,\text{SW}}$;

$$n_{e,\text{SW}} = \frac{(2\pi f_{\text{plasma}})^2 m_e \epsilon_0}{q_e^2} \quad (11)$$

where m_e is the electron mass, ϵ_0 is vacuum permittivity and q_e is the electron charge. f_{plasma} is the corresponding frequency of the plasma line shown in Fig. 5. By introducing a scaling factor η , the distribution function f_{old} can be further calibrated according to;

$$n_{e,\text{SW}} = \eta \int f_{\text{old}} d^3\mathbf{v} \quad (12)$$

in order for the distribution function to correspond to the correct density.

D. Shock parameters

After the corrections have been made to the distribution function, all the shock parameters and the entropy are calculated. All the upstream and downstream quantities used to compute the shock parameters are a 45 seconds mean in the specified region. Upstream electron plasma beta β_e , shock angle θ_{Bn} , whistler Mach number and the solar wind temperature anisotropy A_e are calculated as equation (4), (5), (6) and (8) respectively, whereas the ion ram pressure, the Alfvén Mach number and the ratios need some further calculations.

In order to compute the Alfvén Mach number (equations (2) and (3)) and the ion ram pressure (equation (7)), the upstream flow velocity $V_{n,u}$, which is the relative speed between the solar wind and the shock, must be determined. Following the same procedure as M. Lindberg et al. [1], the solar wind velocity is obtained from the MMS instruments whereas the shock velocity is calculated as the mean of the mixed methods of both the mass flux method and the Smith Burton method, described in [18]. Due to the mass flux method producing physically incorrect results when θ_{Bn} is close to 90 degrees as stated in [1] and [18], only the Smith Burton method is used for those cases.

The ratios in density n_e and temperature T_e are calculated as,

$$\frac{\Delta x}{x_{sw}} = \frac{x_{MS} - x_{sw}}{x_{sw}} \quad (13)$$

where x is a variable to display the equation for both cases. Due to lack of plasma frequency measurements in the magnetosheath, the ion density is used instead of the electron density, since they essentially are the same because of quasi-neutrality in plasma. For the entropy change, equation (9) is used in the magnetosheath and in the solar wind.

IV. RESULTS AND DISCUSSION

The goal of the project was to find shock crossings of the right conditions, regarding date, shock angle and β_e , and analyze these for different parameters and their impact on the change in entropy. As a result, a database of 13 shock crossings was created, with eight new shock crossings together with five from M. Lindberg et al. [1], based on the 2803 shock crossings in the A. Laiti database [14]. The crossings are displayed in table I.

These 13 shocks occurred between November 2016 and October 2018, meaning that the entire time interval from June 2016 to December 2020 never was used. As a result, it is highly possible that more quasi-perpendicular shock crossings with a low β_e could be found inside the original time interval.

Out of the 2803 crossings, only 467 were quasi-perpendicular shock crossings occurring inside the original time interval. Due to an extensive execution time, the calculation of beta was done in sets, resulting in the investigation of approximately 150 shocks. After investigating these 150 shocks, it could be concluded that shocks with $\beta_e < 1$ are rare since only 8 was found.

When observing figures containing the result, it is clear that both Fig. 7 and 9 indicate a strong impact on the change in electron entropy. The dependence is nearly proportional where a large M_A or ΔT_e indicates a large ΔS_e . In addition to this, in Fig. 8 the relation between the entropy change is almost proportional as well, apart from one stray data point. For both ΔT_e and Δn_e these results were expected due to similar results in [1]. However, the relation between β_e and the change in entropy does not seem to be as clear when only plotting $\beta_e < 1$ in Fig. 6, oppose to the strong correlation seen in [1].

Furthermore, the connection between M_A and ΔS_e was later used to group together shocks with similar β_e correlating to similar M_A . These groups were used in Fig. 13 and 14. Regarding Fig. 13, the grouping was to be compared with theoretical

TABLE I
THE SHOCK CROSSINGS AND THEIR QUANTITIES

Crossing	β_e	M_A	θ_{Bn}	$B_{sw}[\text{nT}]$	$\Delta n/n_{sw}$	$\Delta T_e/T_{e,sw}$	$V_{sw}[\text{km/s}]$	M_A/M_{wh}	$P_{i,ram}[\text{nPa}]$	A_e	$\Delta S_e/k_B$
1. 2016-12-09 10:29	0.60	10.9	85	7.9	2.9	5.3	617	5.38	2.05	-0.22	1.44
2. 2017-01-18 05:39	0.50	4.8	65	17.1	1.9	2.5	374	5.38	2.05	-0.22	0.75
3. 2017-01-31 10:07	0.90	10.3	71	9.1	2.5	4.4	645	1.45	3.48	-0.11	1.27
4. 2017-11-02 04:27	0.90	4.7	63	9.9	1.7	1.6	317	0.48	0.67	0.02	0.27
5. 2017-11-24 23:20	0.40	4.4	83	9.1	2.1	3.6	396	1.61	0.58	-0.13	1.30
6. 2018-04-20 09:01	0.35	4.5	83	17.9	0.8	1.3	457	1.7	1.79	-0.27	0.62
7. 2018-04-20 08:05	0.31	2.9	65	20.4	1.6	0.8	423	0.31	1.4	-0.24	0.12
8. 2018-04-20 08:13	0.30	3.3	88	19.4	1.6	1.2	425	5.76	1.79	-0.09	0.20
9. 2016-11-12 12:31	0.58	7.8	62	8.9	3.2	4.1	652	0.77	1.9	0.0005	1.51
10. 2017-01-04 05:34	0.90	8.3	57	14.9	1.4	2.5	399	0.72	3.65	-0.01	1.06
11. 2016-11-12 12:29	0.58	5.2	75	8.9	3.1	1.9	652	0.94	1.1	-0.21	0.43
12. 2018-04-20 09:08	0.58	4.3	77	16.1	1.8	1.1	471	0.89	2.46	-0.08	0.20
13. 2018-10-13 16:12	0.70	4.3	67	11.4	1.5	1.3	388	0.52	0.9	-0.01	0.28

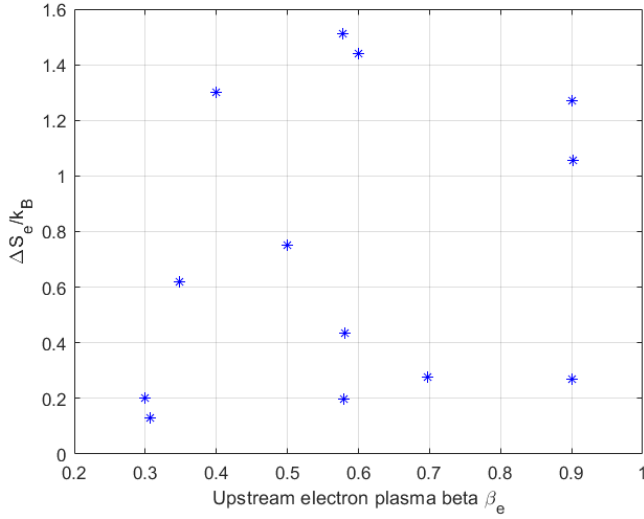


Fig. 6. The relation between the upstream electron plasma beta, β_e , and the change in electron entropy.

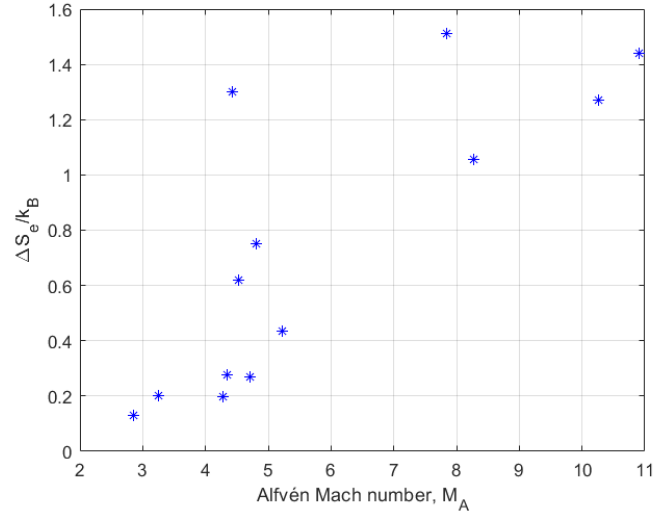


Fig. 7. The correlation between the Alfvén Mach number, M_A , and the change in electron entropy.

predictions produced by M. Lindberg [1]. The predictions, shown in Fig. 15, assumes ideal conditions in which the plasma is considered a 1D fluid, considering both ions and electrons, meaning that ΔS is not the same as our ΔS_e . Fig. 15b was not applied in this study because of the narrow interval of β_e . Regarding M_A , in Fig. 15c, the theoretical and actual values correspond well. Although, the theoretical prediction for θ_{Bn} , in Fig. 15a, is not consistent with the measured and calculated values from the MMS data, seen in Fig. 13. This can on one hand be due to the simplified model under a 1D fluid assumption, and the discrepancy between ΔS and ΔS_e . On the other hand it can also indicate a stronger dependence on M_A and β_e than assumed to begin with. For the blue group in Fig. 13 there is large variation in change in entropy, even though the range of M_A and β_e within the groups are quite small. This could lead to the conclusion that the smaller M_A and β_e , the greater impact on the change in entropy. In addition to this, when observing Fig. 7 it can be seen that at $M_A \approx 4.5$ the change in entropy is between 0.2 and 1.3, which is inside the blue group in Fig. 13. All of these possible explanations are difficult to verify due to the low count of data points. An example of this is the red group in Fig. 13 and 14 which only gives two shocks to base an assumption on.

In Fig. 14 the same groups were used as in Fig. 13. Although there is no graph to compare with to theoretical values, it seems that a dependence close to the one displayed in Fig. 15b can be seen.

Regarding the rest of the graphs in Fig. 10, 11 and 12, no distinct relation can be seen from the data points. It is difficult to say whether this is dependent on the relatively small amount of data or if the lack of relations can be concluded.

V. CONCLUSION

The aim of this project was to examine quasi-perpendicular

shock crossings across Earth's bow shock, in order to find correlations between shock parameters and the change in electron kinetic entropy. This study is a continuation of [1], where it was discovered that a low electron plasma beta indicate a large change in entropy. Using the above mentioned paper and it's methods as a base, we were able to analyze 13 shock crossings of a low β_e and find an additional dependence on the Alfvén Mach number. Furthermore, the dependence of the change in electron temperature and electron density, already discovered in [1], was further strengthened by the results of this study. Regarding future research, it would be interesting to further investigate the dependence on other shock parameters by using limited ranges of both β_e and M_A . This would require larger amounts of data in order to generate a reliable result.

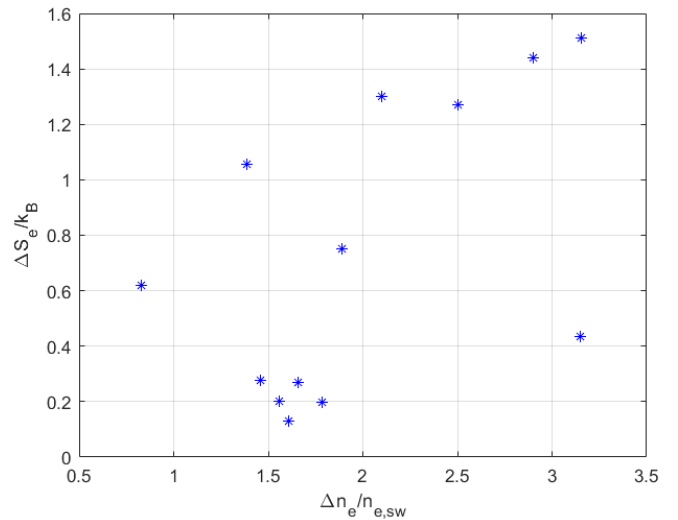


Fig. 8. The relation between the change in electron density and the change in electron entropy.

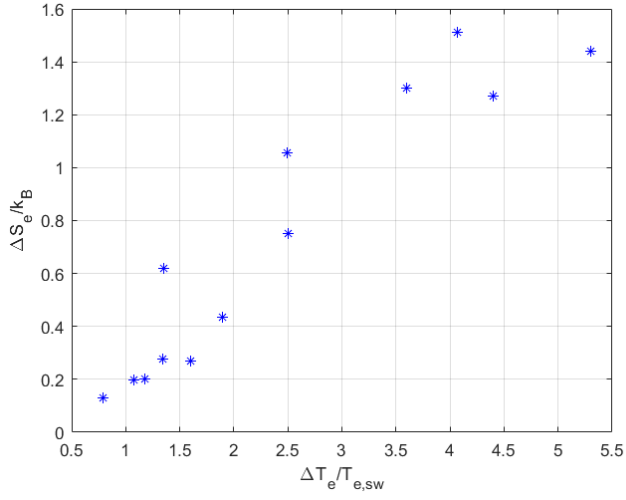


Fig. 9. The relation between the change in electron temperature and the change in electron entropy.

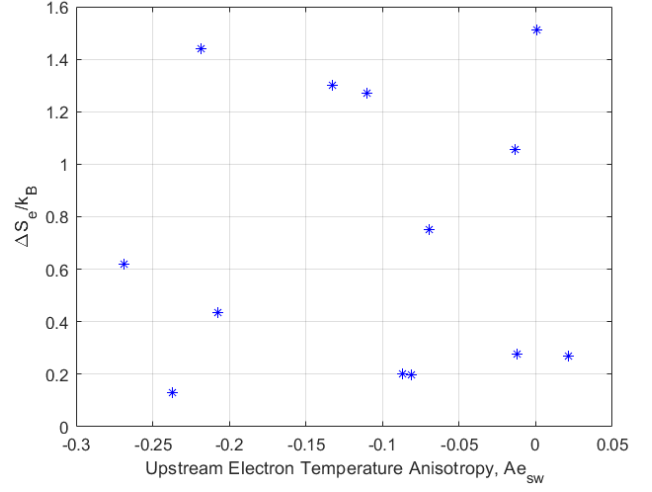


Fig. 12. The dependence of the electron temperature anisotropy, A_e , and the change in electron entropy.

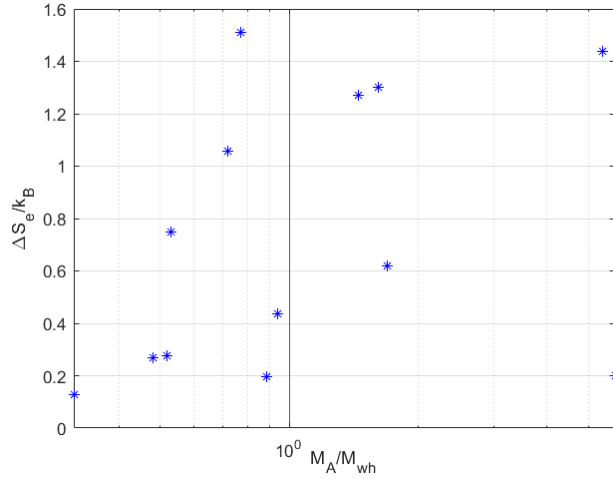


Fig. 10. The ratio of Alfvén to whistler Mach number related to the change in electron entropy.

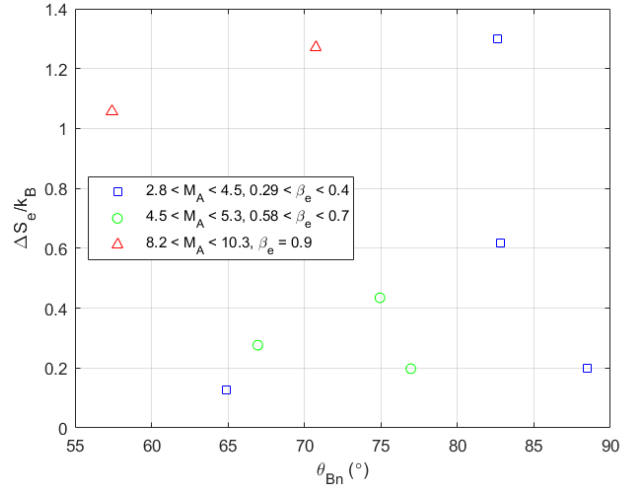


Fig. 13. The connection between the shock angle, θ_{Bn} , and the change in entropy, with fixed groups of M_A and β_e .

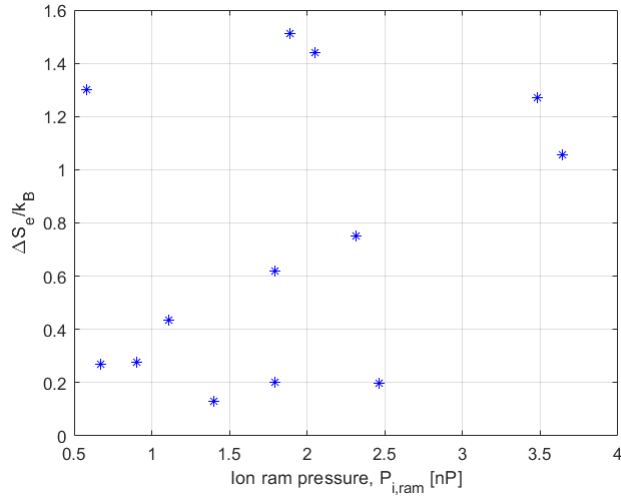


Fig. 11. The relation between the ion ram pressure and the change in entropy.

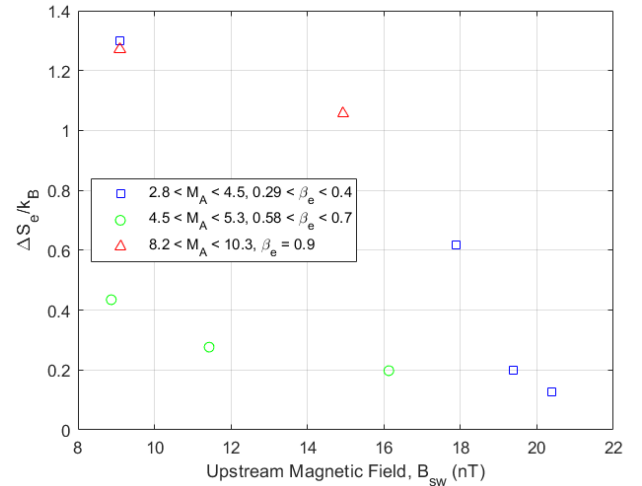
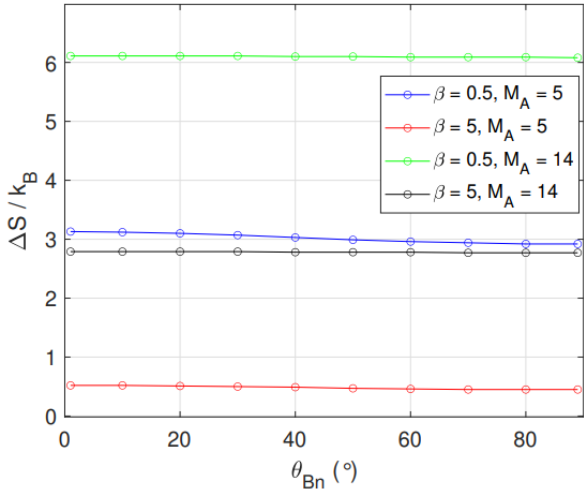
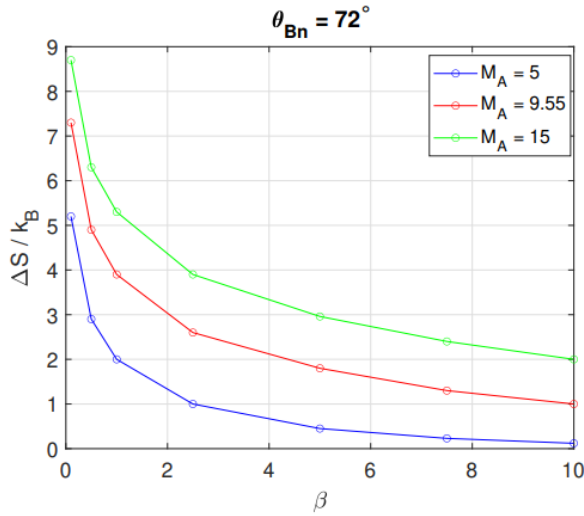


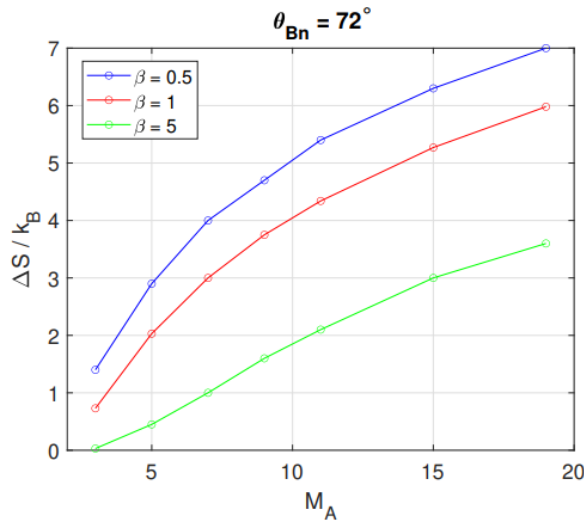
Fig. 14. The connection between the upstream magnetic field, B_{sw} , and the change in electron entropy.



(a)



(b)



(c)

Fig. 15. Theoretical predictions and approximations of the dependence of (a) θ_{Bn} , (b) total plasma beta β and (c) M_A related to the change in total entropy for a 1D fluid plasma. Relations obtained from [1].

ACKNOWLEDGMENT

The authors would like to thank our ever so encouraging and available supervisor Martin Lindberg. We would also like to thank Andris Vaivads, Jan Karlsson and Ahmad Lalti for their partaking in making this study viable.

REFERENCES

- [1] M. Lindberg, A. Vaivads, S. Raptis, P.-A. Lindqvist, B. L. Giles, and D. J. Gershman, "Electron kinetic entropy across quasi-perpendicular shocks," *Entropy*, vol. 24, no. 6, p. 745, Apr. 2022.
- [2] G. K. Parks, E. Lee, M. McCarthy, M. Goldstein, S. Y. Fu, J. B. Cao, P. Canu, N. Lin, M. Wilber, I. Dandouras, H. Réme, and A. Fazakerley, "Entropy generation across earth's collisionless bow shock," *Phys. Rev. Lett.*, vol. 108, p. 061102, Feb. 2012. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.108.061102>
- [3] D. A. Tidman and N. A. Krall, *Shock waves in collisionless plasmas*. Hoboken, NJ: John Wiley & Sons, 1971.
- [4] (2015, Mar.) Magnetospheric multiscale: Using earth's magnetosphere as a laboratory to study the microphysics of magnetic reconnection. National Aeronautics and Space Administration. Press kit. [Online]. Available: https://www.nasa.gov/sites/default/files/files/MMS_PressKit.pdf
- [5] C. Kennel, J. Edmiston, and T. Hada, "A quarter century of collisionless shock research," *Washington DC American Geophysical Union Geophysical Monograph Series*, vol. 34, pp. 1–36, 1985.
- [6] A. Lalti, Y. Khotyaintsev, D. B. Graham, A. Vaivads, K. Steinvall, and C. T. Russell, "Whistler waves in the foot of quasi-perpendicular supercritical shocks," *arXiv preprint arXiv:2011.10593*, Nov. 2020.
- [7] M. Oka, T. Terasawa, Y. Seki, M. Fujimoto, Y. Kasaba, H. Kojima, I. Shinohara, H. Matsui, H. Matsumoto, Y. Saito *et al.*, "Whistler critical mach number and electron acceleration at the bow shock: Geotail observation," *Geophysical Research Letters*, vol. 33, no. 24, Dec. 2006.
- [8] G. W. F. Drake. (2021, Jun.) Thermodynamics. Encyclopedia Britannica. [Online]. Available: <https://www.britannica.com/science/thermodynamics>
- [9] H. Liang, P. A. Cassak, S. Servidio, M. A. Shay, J. F. Drake, M. Swisdak, M. R. Argall, J. C. Dorelli, E. E. Scime, W. H. Matthaeus *et al.*, "Decomposition of plasma kinetic entropy into position and velocity space and the use of kinetic entropy in particle-in-cell simulations," *Physics of Plasmas*, vol. 26, no. 8, p. 082903, Apr. 2019.
- [10] C. Russell, B. Anderson, W. Baumjohann, K. Bromund, D. Dearborn, D. Fischer, G. Le, H. Leinweber, D. Leneman, W. Magnes *et al.*, "The magnetospheric multiscale magnetometers," *Space Science Reviews*, vol. 199, no. 1, pp. 189–256, Mar. 2016.
- [11] C. Pollock, T. Moore, A. Jacques, J. Burch, U. Gliese, Y. Saito, T. Omoto, L. Avannov, A. Barrie, V. Coffey *et al.*, "Fast plasma investigation for magnetospheric multiscale," *Space Science Reviews*, vol. 199, no. 1, pp. 331–406, Mar. 2016.
- [12] P.-A. Lindqvist, G. Olsson, R. Torbert, B. King, M. Granoff, D. Rau, G. Needell, S. Turco, I. Dors, P. Beckman *et al.*, "The spin-plane double probe electric field instrument for MMS," *Space Science Reviews*, vol. 199, no. 1, pp. 137–165, Mar. 2016.
- [13] R. Ergun, S. Tucker, J. Westfall, K. Goodrich, D. Malaspina, D. Summers, J. Wallace, M. Karlsson, J. Mack, N. Brennan *et al.*, "The axial double probe and fields signal processing for the mms mission," *Space Science Reviews*, vol. 199, no. 1, pp. 167–188, Mar. 2016.
- [14] A. Lalti, Y. V. Khotyaintsev, A. Dimmock, A. Johlander, D. Graham, and V. Olshevsky, "A database of MMS bow shock crossings compiled using machine learning," *arXiv preprint arXiv:2203.04680*, Mar. 2022.
- [15] MATLAB, version 9.9.0 (R2020b). Natick, Massachusetts: The MathWorks Inc., 2020.
- [16] (2021, Sep.) IRFU-Matlab analysis package. [Online]. Available: <https://github.com/irfu/irfu-matlab>
- [17] D. J. Gershman, L. A. Avannov, S. A. Boardsen, J. C. Dorelli, U. Gliese, A. C. Barrie, C. Schiff, W. R. Paterson, R. B. Torbert, B. L. Giles *et al.*, "Spacecraft and instrument photoelectrons measured by the dual electron spectrometers on MMS," *Journal of Geophysical Research: Space Physics*, vol. 122, no. 11, pp. 11–548, Oct. 2017.
- [18] G. Paschmann and P. W. Daly, "Analysis methods for multi-spacecraft data. issi scientific reports series sr-001, esa/issi, vol. 1. isbn 1608-280x, 1998," *ISSI Scientific Reports Series*, vol. 1, 1998.

CONTEXT L

OBSERVATION PLATFORMS AND INSTRUMENTATION FOR SPACE PHYSICS

POPULAR SCIENCE

Are we alone? No, we are surrounded by elves and trolls!

Elves, trolls, and gnomes sound like something straight out of a fairy tale, but a lucky observer might be able to spot one if they look high enough. Indeed, scientists with a sense of humour have dubbed intriguing electromagnetic events after these mystical creatures. These electromagnetic phenomena, like upper-atmospheric lightning and northern lights, dance elusively hundreds of kilometres above our heads. In addition to being beautiful, these phenomena are related to the weather that affects us every day.

Exploring the fantastic world that exists above us requires awesome vehicles like high-altitude autonomous planes and satellites. Developing and improving them will not only give a unique insight into the amazing light shows performed in the sky, but also improve weather predictions and climate research. Once this realm has been accessed, a more intimate relationship with even more electromagnetic events can blossom.

Students at KTH Royal Institute of Technology are building different observational platforms in order to study these near-Earth plasma phenomena closer. An unmanned drone (ALPHA UAV) is being developed will fly high up in the atmosphere in order to photograph these electromagnetic events that we cannot easily see from the surface. Going even higher, the student satellite teams (MIST satellite and REXUS rocket experiment) will be launching their platforms to space and fly through the northern lights and thunderstorms in the upper atmosphere, taking measurements of the magnetic field.

Having a better way to observe these events in our atmosphere could help us better understand the atmosphere of other planets or moons. These platforms could pave the way for further research on other worlds and assist our civilization in its pursuit to become multi-planetary.

SUMMARY OF PROJECT RESULTS

In order to better understand our home planet, we need to also understand what is happening around us. Near-Earth space is heavily trafficked by navigation and telecommunication satellites, space stations, and telescopes, which is why an understanding of this environment is important. Observational platforms are necessary to study this region and these vehicles need accurate instrumentation to provide useful data on the earth below us, the sky above us, and the space beyond. With the increasing number of satellites, it is important to ensure that the vehicles are capable of controlling themselves autonomously, to avoid proximity risks. By gathering data we open the road to creating bigger datasets and thereby build reliable models in all of these fields.

By using both data from low Earth orbit satellites and stratospheric data from unmanned aerial vehicles (UAV) we can observe atmospheric phenomena from more perspectives. The instrumentation has a variety of functions ranging from satellites measuring the magnetic field, to taking pictures of upper atmospheric and space phenomena.

Various student projects focusing on space observation are currently being conducted at KTH Royal Institute of Technology. One of them is the Autonomous Light Platform for High Altitude UAV (ALPHA) project, whose mission is to take pictures of upper-atmospheric phenomena. The Miniature Student satellite (MIST) project is about building a nanosatellite that will be

the main platform for new technological solutions from the industry and KTH's own research. KTH also has a long history in the context of the REXUS/BEXUS (Rocket and Balloon Experiments for University Students) program, which is a bilateral Agency Agreement between the German Aerospace Center (DLR) and the Swedish National Space Agency (SNSA). Students are given the opportunity to launch their own experiments on research rockets and balloons.

One of the low Earth orbit instrument projects is **project L1**. The group's task is calibrating the MIST magnetometer. The magnetometer is needed to determine the attitude and orientation of the satellite, as the MIST satellite relies on the magnetic field of the Earth as a reference. The magnetometer needs to measure the magnetic field without the magnetic disturbances that are produced by the other systems in the satellite. The calibration process can be divided into phases. In the first phase, group L1 measures the magnetic field with the magnetometer in an environment where we can control the magnetic field applied to it, in our case we use Helmholtz coils. By comparing the applied magnetic field to the measurements we can determine the offsets in the magnetometer. In the second phase, the L1 group writes a MATLAB script that will use a mathematical model to correct the magnetic disturbances from the satellite itself. This is done to ensure that the measured field matches the real one. The end result is a set of calibration parameters (scales and offsets) that will convert the magnetometer measurements to accurate measurements of the Earth's magnetic field. This gives a good picture of the satellite's orientation in real-time.

Project group L2 aims to understand the effects of space weather on communications and navigation systems. The group focuses mostly on a wave propagation experiment that aims to be implemented in a miniaturised payload that was developed for the REXUS PRIME experiment flown in 2019. In this project, the focus is on choosing the electronics for the acquisition of the signal on the payload and providing a solution that can be implemented in future experiments. Particular attention is put into designing the system to be robust against noise and being able to reconstruct the signal in the receiver. In the second phase, the work is moved to developing the software written in hardware descriptive language for Field Programmable Gate Array (FPGA). Lastly, we aim to assemble all the components and test them in the implementation phase, where the functionality of the experiment is tested. The results from project L2 open the door for this kind of research to be implemented on small new-generation rockets.

Project group L4a aims to create a minimum value product (MVP) of the entire electrical system that is needed for a working autonomous flying aircraft. This system will lay the groundwork for the electrical system on the ALPHA UAV. The electrical system consists of power electronics, flight electronics, and a radiolink. The work on power electronics is focused on finding and implementing batteries, engines, and engine drivers for the propulsion of the UAV. Flight electronics consists of sensors and computing hardware to gather necessary data and do necessary calculations to control engines and control surfaces to enable autonomous flight. The radio link must be enabled for the operator to monitor and if necessary control the UAV. The project group also aims to test the electrical system on a UAV during test flights. The purpose of these flight tests is to evaluate the system and make further adjustments.

Project group L4b focuses on how to develop the proportional integral derivative (PID) controller for the control surface that controls the pitch motion by using data from flight tests with a UAV. Finding a suitable method to develop the controllers is essential to make the UAV autonomous, whilst being stable and controllable. The research in this project focuses on evaluating a method to model the response from the control surface and thereafter finding the PID for the control surface. Before developing the PID, data from test flights are used to model a transfer function for the control surface by using system identification with MATLAB. Thereafter, using the transfer function, the PID is developed by using MATLAB's toolbox for control systems. The whole method is evaluated by studying the rise time, settling time, and overshoot for the PID, and studying how well the transfer function fits with the flight data.

In order to make the ALPHA UAV fly and remain at the altitude, position, and speed specified for the missions, a propulsion system is required. **Project group L5** is studying the implementation of an electric propulsion system for ALPHA. The focus is on electric motors and electronic speed controllers for the autopilot to interface with the motors. By testing a variety of electric motors, propellers, and speed controllers, specifications are derived in order to fulfil the requirements of the drone. In conjunction with a study of available and feasible options, optimal decisions on the propulsion system can be made and implemented.

The results of these projects will be able to improve near-Earth space observations as a whole. For example, when observing upper atmospheric phenomena, a controllable observational satellite can provide images from above whereas a UAV will be able to provide images from below a rocket-based experiment from within. This will give a more complete picture of these phenomena and aid in atmospheric research.

Further development in this field could entail propagating the findings for satellites and aerial vehicles to other platforms for Earth and space exploration, like sounding rockets, balloon flights, and observational platforms on the International Space Station. The ALPHA project is a continuing project which could benefit from added instrumentation and expansion of the flight envelope. Making autonomous systems reliable would open the door for further exploration on faraway planets where direct control of vehicles is not possible. The calibration of the magnetometer on-board the MIST satellite will be used for future calibration processes as a reference.

IMPACT ON SOCIETY AND ENVIRONMENT

The project groups in context L focus on developing platforms that will contribute to space observations. These types of platforms can be used both for good and potentially bad purposes, depending on the intentions of the user. On the one hand, continuously developing better technologies for satellites allows us to improve data sharing and collecting, which is an integral part of today's society. Vast amounts of research are done in these areas to improve for example weather forecasting, which relies on collecting data with the help of platforms like satellites.

On the other hand, observational platforms can also be used for military and espionage purposes. When conducting studies using vehicles capable of observation, there exists a risk of provoking geopolitical tension if an unintended path or trajectory is taken. This could happen in the case of communication loss or hacking attacks, which could cause unintended tracking and espionage. However national laws and permits are in place to minimise these risks and the value of research and data has to be weighed against the risk of misuse of the equipment. The location of these observations should be chosen carefully to avoid proximity to hostile countries, where they could be misinterpreted as espionage. In the case of REXUS, the rocket launch could be misinterpreted as a hostile missile launch, which is why this year's REXUS launch was unfortunately cancelled.

GPS and other space systems affect almost all individuals in their daily lives. Even though we don't see a clear impact right now, it's still relevant to invest in the research of platforms such as ALPHA and MIST. These kinds of instruments will lead to the collection of great amounts of data, from images to other kinds of information, and enable improved communication between users. Our purpose as engineers is to create the platforms with the original goal of space exploration and observation in mind. It would then be up to policy- and lawmakers to make sure that these are used in non-harming ways and that laws are put in place to counteract such usages. The responsibility for using the projects ethically is thus shared between engineers such as ourselves and lawmakers.

Conducting observations and research in space can lead to the build-up of space debris, which is a growing problem. This leads to the dilemma of whether the conducted missions justify the added toll on the near-Earth environment. Utilising space for atmospheric observations can make important contributions to research areas such as climate change and severe weather phenomena. As of now climate change around the world is largely regarded as a more pressing issue than the one of space debris and therefore a consensus is reached that Earth-observation satellites are well justified. The overall impact of Earth-orbiting satellites on climate is however not at all as big as e.g., aviation, as there are rules in place to limit the effects on climate. For example, the requirement for normal satellites to be deorbited within 25 years, is mentioned in the 'space code'.

The ALPHA UAV will consist of a lot of non-renewable materials including batteries and composites. Non-renewable is inherently bad for the environment. It is also questionable whether electric propulsion in this application is the most environmentally friendly option. Batteries and electric motors also use materials that might be unethically sourced. In the case that control of the drone is lost; it will fall uncontrollably under a parachute until it impacts an unknown landing location where it might never be recovered. This could leave composite materials, plastics, batteries, and other equipment in the wilderness or in the ocean. Aside from the ecological damage that a loss of control could cause, there is also possible property damage or even personal injuries that should be taken into account. Developing autonomous flight should be done with

safety in mind: what happens when you lose control of the aircraft? How do you reduce the possible damage from a crash? From an individual standpoint, it is dangerous to be crashed into by an aircraft, due to the high velocity of the object, its mass, and possible cutting edges such as the propellers. Developing robust navigation systems and autopilots is necessary to mitigate the risk of crashes and ensure that there is no loss of control. Context L works mainly on small UAVs and satellites that are or will be extensively tested before being sent into space or the atmosphere. The probability of significant personal injury or property damage is low enough that the project members of context L agree that it is justifiable to conduct the research.

As with any venture, be it science or other, risks are involved. However, the benefits of improving platforms for space and atmospheric observations are many. With the ongoing development of the technology studied in the projects of this context, accelerated actions against climate change can be taken and risks associated with severe weather phenomena can be avoided. Most of all it will better our understanding of the planet we live on and the environment that surrounds it.

Characterisation of Satellite Onboard Magnetometer for MIST

Marcus Mhanna

Abstract—The most common equipment used for attitude determination in small satellites are magnetometers. However, using magnetometers gives rise to many challenges. One of these challenges is the calibration of the magnetometer. Magnetometer calibration takes many factors into account. There are external and internal factors. External factors can be the satellite itself. Satellites are built of many complex subsystems. These subsystems can produce magnetic disturbances and affect the measurements taken by the magnetometer, which also affects the attitude determination of the satellite. Internal factors are non-orthogonality and scale factors. In this project, we aim to test different calibration methods and compare the results. Another objective is to provide a complete procedure for a calibration of the magnetometer using the Helmholtz coils. The comparison of the results with other methods can help with the decision of which should be used to calibrate the magnetometer onboard the satellite for future calibrations for MIST satellites.

Sammanfattning—En av de mest vanliga verktyg för attitydbestämning i små satelliter är magnetometer, men att använda magnetometer kan leda till många utmaningar, en av de är kalibrering av magnetometern. Magnetometer kalibrering är beroende av många faktorer. Det finns inre och yttre faktorer. Yttre faktorer kan vara själva satelliten. Del system som bildar satelliten kan påverka mätningar i magnetometern och då påverkar attitydbestämning av hela satelliten. Inre faktorer är icke ortogonalitet och skalära faktorer. I det här projektet vi ska testa olika kalibrerings metoder och jämföra resultaten. Ett annat mål är att bygga en komplett procedur för att kalibrera magnetometer med hjälp av Helmholtz spolar. Jämförelsen och resultaten från kalibreringen visar hur det är möjligt att kalibrera en magnetometer som är integrerad i satelliten för kommande kalibreringar i MIST.

Index Terms—Magnetometer calibration, Ellipsoid Fitting, Helmholtz coil

Supervisors: Nickolay Ivchenko, Sven Grahn

TRITA number: TRITA-EECS-EX-2022:157

I. INTRODUCTION

Over the past decade, a big interest for low budget satellites has arisen especially with the advancements in telecommunication systems and the internet. Many companies have shown interest in covering the planet with internet so they can reach larger markets and to make the internet more accessible. Besides the commercial use, the research aspects such as space observation platforms have become more and more important to study space phenomena. Nano cubesat satellites have proven to be good for low budget space studies satellites and one of the projects on that topic is MIST in KTH space center. MIST or Miniature Student saTellite is a student project aimed to help the students work on satellites and help them improve their knowledge about space related projects. The current MIST project is building a 3U cubesat satellite [1], which means

here the satellite is low orbit and will use the geomagnetic field of the earth as reference for attitude determination.

The geomagnetic field of the earth is disturbed by electric currents associated with aurora. The disturbances can be high as a few percents of the internal field magnitude and can cause attitude determination errors. As stated before, these satellites often used for space observations and it is important to get accurate measurements of the magnetic field as it contains information on aurora currents which can be of interests in space physics research. To get a precise attitude determination system, an accurate calibration of the magnetometer is required [2].

In the case of the MIST satellite, it will use off the shelf magnetic sensor that will be integrated in the satellite. The reading from the magnetic sensor must be calibrated and the disturbances caused by the satellite itself must be removed. The calibration process is mainly comparing a known magnetic field vector with measured magnetic field from the sensor. The geomagnetic field can be used in this process by rotating the sensor in a place where the geomagnetic field is known and with the least magnetic disturbances possible in order to get an accurate calibration, such place can be a forest.

Another method is to use Helmholtz coils by generating magnetic field on different angles of the sensor and placing the sensor inside the set of coils [3]. Using Helmholtz coils provides easier and more accurate calibration as we can control and apply different magnitudes of magnetic fields.

The goal of this study is to have a calibrated magnetic sensor and to provide an accurate calibration process that will help current and future projects at KTH. Two sensors were used in this study, the first is Small Magnetometer In Low-mass Experiment (SMILE) sensor and the other one is IMTQ. The first sensor will be used mainly in this study but data will be collected from the IMTQ as it will be used in the real satellite and any risks of damaging it should be avoided. The data from both magnetometers is analyzed and studied as one will be used for testing and the other one will be used on the MIST satellite project.

II. MATHEMATICAL FORMULATION AND METHOD ANALYSIS

The magnetometer calibration is a configuration of the magnetometer that makes it provide measurements within a more accurate range by eliminating or minimizing factors that can cause inaccurate measurements [4]. Some of the factors are offsets caused by the production process. External factors can be disturbances and magnetic current from the satellite itself which can affect the measurements taken by the magnetometer.

In this study we will focus mainly on the internal factors and fix them.

A. Magnetometers

Magnetometers are devices that are used to measure the magnetic field. There are many types of magnetometers that measures the magnetic field in different ways. For example scalar magnetometers measures the strength of the magnetic field while vector magnetometers measure the magnetic field on three axes orthogonal to one another. The magnitude of that magnetic field can be calculated by taking the square root of the sum of squares of these components. Both of the magnetometers that were used in this project are vector magnetometers.

The first magnetometer that was used for testing and analyzing the different methods is the SMILE sensor shown in Fig.1.

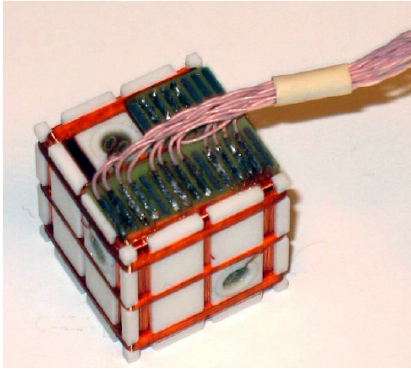


Fig. 1: SMILE sensor [5]

It is a fluxgate magnetometer [5]. Fluxgate magnetometers consists of a core made of material with high magnetic susceptibility and two coils around it. The coil has a supply of alternating current and with a changing field; it induces electric current in the second coil. The difference between the input current flowing through the first coil and the second one will be dependent on the magnetic field in the background. The second magnetometer that was used in the project is IMTQ shown in Fig.2.

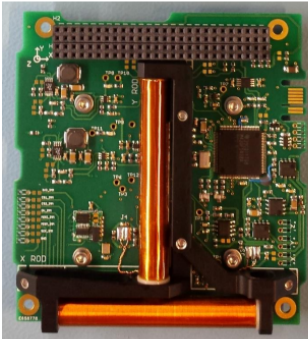


Fig. 2: IMTQ

IMTQ uses XEN1210 magnetometer to measure the magnetic

field on three different axis. XEN1210 uses the Hall effect to measure the magnetic field. Electric current run through a semiconducting material and the magnetometer XEN1210 assess the distortion in the current due to the magnetic field in the background. The voltage where that distortion occurs is called the Hall voltage and it is proportional to the magnetic field [6].

B. Scaling factors

Both magnetometers that were used in this project are vector magnetometers and they measure the magnetic field in three directions. Three magnetic sensors are placed in three axes orthogonal to one another. These three axes are meant to measure in three dimensions. The three magnetic sensors are supposed to sense the magnetic field in the same way, by applying the same magnitude of the magnetic field as an input, the expected output should be the same; however due to flaws that are common in these types of designs, it becomes a need to scale the outputs of the different sensors so we get the same measurements from the sensors. It is important to analyze and fix the scaling factors because the the main function of the magnetometer is attitude determination. Any inaccuracies in the the measurements in one of the sensors can provide false data that can be hard to fix or calibrate when the satellite is in orbit. To represent the scale factors on three axes, we get a matrix with an offset factors in the diagonal representing the scale factor in the three axes as shown in the following matrix:

$$C_s = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \quad (1)$$

C. Offsets

The offsets are the measured magnetic field in the null field. The offsets are biases in the magnetic field that could be the results of many things. To identify these offsets that can come in all three axes, it can be represented by the 3x1 vector B_{offset} with each element representing the the offset bias in an axis shown in the following vector:

$$B_{offset} = \begin{bmatrix} off_x \\ off_y \\ off_z \end{bmatrix} \quad (2)$$

D. Non-orthogonality

Other important factor that can be product of mass production is non-orthogonality between the magnetic sensors. Non-orthogonality can cause false measurements as one magnetic sensor with an angle can measure magnetic fields from the other two axes.

For this problem a transformation matrix Co can be used to adjust the axes in the measurements. The method for determining the matrix is taking one axis as a reference and transforming the other two axes to be orthogonal towards the axis and each other. In Fig.3, the reference is the z' axis and the y' axis has an angle with the zy plane defined as φ . The angle between the x' and xz plane is defined as θ and the

angle between x and y is defined as ϕ as shown in the next matrix:

$$C_o = \begin{bmatrix} \sin(\theta) * \sin(\varphi) & \cos(\phi) * \sin(\theta) & \cos(\theta) * \sin(\phi) \\ 0 & \sin(\varphi) & \cos(\varphi) \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

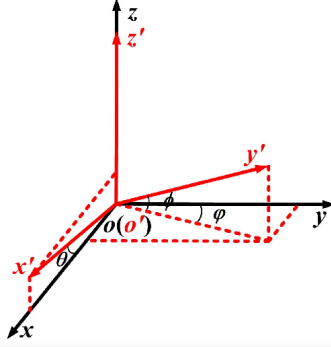


Fig. 3: Demonstration of non-Orthogonality [7]

E. Misalignment

The magnetometers are mainly used for attitude determination in satellites. The magnetometers not being aligned with the satellite's axes can cause false measurements thus provide false attitude determination. It is considered one of the external factors as it occurs mostly due to errors in integration of the magnetometer on the satellite.

F. Mathematical Model

After identifying all the factors that needs to be analyzed, the mathematical model for the magnetometer calibration can be given as:

$$B_{real} = C^{-1}(B_{measured} - B_{offset}) \quad (4)$$

There B_{real} is the real magnetic field. C is the correction matrix with the scaling factors in the diagonal (1). B_{offset} is the offsets in the measurements of the null field 2). $B_{measured}$ is the measured magnetic field from the magnetometer.

G. Ellipsoid fitting

The concept of ellipsoid fitting is that the magnetometer taking measurements in one point should give the same magnitude independently of the orientation. When we plot the measurements we should get a sphere as a representation of an ideal magnetic field measurement that has constant magnitude not dependent on the orientation. For uncalibrated magnetometers, the offsets in sensors and non-orthogonality will affect the measurements and not provide sphere shaped plot when rotating and measuring the magnetic field. As shown in Fig.4, the blue points represent uncalibrated measurements and the red points are the calibrated measurements and the sphere is the ideal magnetic field measurement. To adjust the

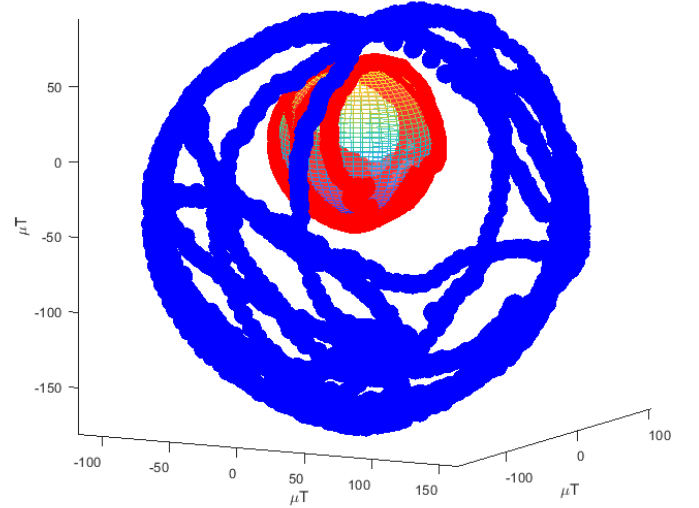


Fig. 4: Sphere representing the ideal magnetic field measurement

blue uncalibrated measurements, we use adjusted least square algorithm. By finding the least square solution between the uncalibrated and ideal measurements and adjust the uncalibrated data with the ideal data. The solution will be in form of scale factors and these scale factors are the elements of matrix S (1) multiplied with the non-orthogonality matrix C_o in (2).

H. Adjusted Least Square

The theory behind the ALS estimator is based on Quadratic measurement error model and Ordinary least square estimator [8]. The quadratic measurement error model taken from [8, eq.(6)] is a second order surface in \mathbb{R}^n with the following set:

$$S(A, b, d) := x \in \mathbb{R}^n : x^T A x + b^T x + d = 0, \quad (5)$$

in which A is a symmetric matrix $A \in \mathbb{S}$. The vector $b \in \mathbb{R}^n$ and scalar $d \in \mathbb{R}$ are the parameters of the surface. The matrix A and vector b defines the shape of that surface. With (5) being non-identifiable, it is resolved by imposing a normalising condition as mentioned in [8, eq.(9)] in the following form:

$$\|\bar{A}\|_F^2 + \|\bar{b}\|^2 + \bar{d}^2 = 1 \quad (6)$$

From (6) and (5), the Ordinary least square estimator is a global minimum point of the optimization problem:

$$\begin{aligned} \min_{a,b,d} \quad & \sum_{l=1}^m (x^{(l)T} A x^{(l)} + b^T x^{(l)} + d^2) \\ \text{s.t.} \quad & A \mapsto \text{Symmetric} \\ & \|\bar{A}\|_F^2 + \|\bar{b}\|^2 + \bar{d}^2 = 1 \end{aligned} \quad (7)$$

but with the OLS estimator being inconsistent, an adjustment procedure is proposed in [9] and according to [8] the ALS estimator $\hat{\beta}_{als}$ is a global minimum point for the next optimization problem:

$$\begin{aligned} \min_{\beta} \quad & Q_{als}(\beta) \\ \text{s.t.} \quad & \|H\beta\|^2 = 1 \end{aligned} \quad (8)$$

where H is a non-singular matrix and the ALS cost function described in [8] is

$$Q_{als}(\beta) = \sum_{l=1}^m q_{als}(\beta; x^{(l)}), \text{ for all } \beta \in \mathbb{R}^{n\beta} \quad (9)$$

A more indepth explanation of ALS can be found in [8].

I. Helmholtz coils

The Helmholtz coils is device that is used to control the magnetic field in a region. It uses electric current to induce magnetic field inside the Helmholtz coils. The Helmholtz coils as a concept is based on Ampere's law with the following formula:

$$\oint_P \vec{B} \cdot d\vec{l} = \mu_0 I_{enc} \quad (10)$$

which is a line integral of the magnetic field B in a closed path P that is proportional to the current enclosed by the path. μ_0 is a constant and it is the permeability of free space and it refers to the rate of magnetization for a material given a magnetizing field. I_{enc} is the current enclosed by path and it can be calculated with the number of turns N in the length L multiplied by the current I in each coil.

There are two types of Helmholtz coils. The first one is circular Helmholtz coils which consists of two circular magnetic coils. The second is Tri-axial Helmholtz coils system that an example of which is shown in Fig.6. The magnetic field B at the centre of tri-axial Helmholtz coils is governed by the following equation according to [10]:

$$B = \frac{2\mu_0 NI}{\pi a} \frac{2}{(1 + \gamma^2)\sqrt{2 + \gamma^2}} \quad (11)$$

there γ is the ratio of the distance between two coils and $2a$ is the length of the side of a coil. Common usages of the Helmholtz coils include magnetometer calibration, magnetic compass calibration and biomagnetic studies [10].

J. Linear curve fitting

Linear curve fitting or also known as linear regression is a data analysis model that finds relationships between two sets of data (x_1, x_2, \dots, x_n) and (y_1, y_2, \dots, y_n) . The (x_1, x_2, \dots, x_n) is the real data and (y_1, y_2, \dots, y_n) is the independent observations of that data which has expected values μ_i are linearly dependent on the data x_i . The theoretical regression line equation :

$$y = \alpha + \beta x \quad (12)$$

which shows the dependency of the expected values μ_i on the regression variable x . β is the coefficient that shows how the expected value changes if x changes and if β is zero then the expected value is constant [11]. The regression line is plotted in a way to fit as many data points (x_i, y_i) and if the data point can not be fitted, it is plotted on the graph in a way to minimize the distance between the line and the data points as much as possible. The most common approach is to use least square method to minimize the distance which represents the error as shown in Fig.5. The mathematical least square model is shown in the following equation:

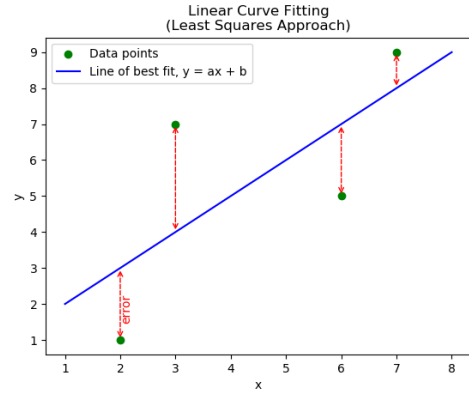


Fig. 5: Linear curve fitting using the least square approach [12]

$$Q(\alpha, \beta) = \sum_{i=1}^n (y_i - \mu_i)^2 \quad (13)$$

there the expected value is $\mu_i = \beta x_i + \alpha$.

III. METHODOLOGY

There are mainly two methods in our approach. Each method uses a mathematical algorithm to determine the calibration matrices. The first one is ellipsoid fitting with adjusted least square (ALS) and the second one is using Helmholtz coils with linear curve fitting (LCF). In total six tests were conducted as shown in Table I.

Table I

The six tests with their locations and methods

Test number	Magnetometer	Location	Method
1	SMILE	MIST Lab	ALS
2	SMILE	MIST Lab	ALS
3	SMILE	Uppsala (Helmholtz coils)	LCF
4	IMTQ	Uppsala (Helmholtz coils)	LCF
5	SMILE	Uppsala (Helmholtz coils)	ALS
6	IMTQ	Uppsala (Helmholtz coils)	ALS

A. Ellipsoid fitting test

The magnetometer has to rotate and take measurements in a point where the magnetic field is known. The main idea is to rotate the magnetometer in a point or location for which the magnetic field is known and then try to minimize the other unknown disturbances in the measurements by doing it in an environment with the least amount of magnetic disturbances possible. In our case, we conducted the first two tests in MIST lab due to not having a good safety measures to avoid damages when moving the magnetometers around at that time. We tried to choose the time in which the least amount of people

are working in the lab. The measurements have been taken and compared to the known magnetic field that is given by IGRF (International Geomagnetic Reference Field) website by inserting our coordinates. The difference will be our biases that needs to be calibrated. The two main offsets that can be solved directly is the non-orthogonality as the transformation method can be included in the software to ensure orthogonality. The scaling factors will be solved using the measurements we took. The magnetometers takes data in three different axes during the rotation and provide enough data points to perform the ALS or Adjusted least square method. A more in-depth explanation of the implementation of the ALS can be found in [8]. Six tests were performed in total and four of them with ellipsoid fitting. The first test is one point measurement in the lab without moving the magnetometer. The test was done in lab and disturbances were expected. The test is mainly to verify that the visualisation and the measurements are plotted in a correct way. The second test is conducted in the same place but with rotation. This test has the same purpose as the first one as well as providing a calibration results that will be compared later on in the discussion. It was expected that it will provide more complicated results. Both of these tests are conducted with SMILE sensor. The fifth and sixth tests were conducted inside the Helmholtz coils with a generated magnetic field of $50 \mu\text{T}$. The magnetometers were rotated in the middle of the Helmholtz coils and measurements were taken so it can be possible to calibrate them using the ALS algorithm.

B. Helmholtz coils

The second calibration method is conducted with Helmholtz coils. In this method we use Tri-axial Helmholtz coils which is magnetic coils positioned to induce magnetic field on three different axis [13]. The magnetometer is set in the middle as shown in Fig. 6. We apply magnetic fields on the three axis of the magnetometer with the help of Helmholtz coils and record how they respond to it. Given the fact the we can control the magnetic field applied on the magnetometer and we know the magnitude it has, we get more accurate results. Then we can test the magnetometer by applying the expected magnetic fields that can occur in the environment surrounding the satellite in space. In that case it will vary between $25 \mu\text{T}$ and $65 \mu\text{T}$. We set the magnetometer on the middle stone inside the coil cage as shown in Fig. 6. Due to the limited time we have to perform the measurements, we try to take as many data points as possible as the experiment costs a lot of money and not possible to do a second time during the period in which the study was conducted in. Some of the equipment that is needed to set the magnetometer was a power bank and a long USB cable. As shown in Fig. 6 the magnetometer has to be far enough from any disturbances. That is why we need a power source that can be far from the magnetometer. In our case we used a USB cable with two meters length which allows us to use a laptop as a power source for the magnetometer. We start with applying the null field on the magnetometer and record the measurements. Then we apply $-60 \mu\text{T}$ by gradually decreasing the magnetic field with step

length of $8 \mu\text{T}$ and time delay of 10 seconds to make sure the magnetometer records it clearly, then we increase with same step length and time delay up to $60 \mu\text{T}$. The choice of going to negative and positive magnetic fields can help with the fitting calibration as we will have more data points to look into. This can also help to determine if the magnetometer has any problems or if it is sensitive to a specific magnetic field. It allows us to check the measurements if they are not linear. Due to us having two magnetometers with two different main functions, it is expected that we get small differences in the measurements.



Fig. 6: Helmholtz coils used

C. Software and Implementation

The SMILE sensor requires converter as it records data in the form ASCII stands for American standard Code for information Interchange . We use the code SMILE Parser that has the algorithm that converts the data acquired by the SMILE sensor to nano Tesla. The code first reads the whole log file that contains the recorded data in the ASCII form as one array shown in Fig.7. In the second step it divides the data into rows and columns forming a matrix. The code checks every 17th element of every row if it is equal to 10 in ASCII table than it is a correct row. If not the code checks every element in that row and removes unexpected characters.

For the IMTQ, the magnetometer records the data in form of matrix with three measurements for the three axes of the magnetic fields and thus does not need to be converted.

3C4A499BD756ABA

3C46C99A6256B7E

3C442999C156C0E

3C473999EE56C19

3C4C999B2656B1E

Fig. 7: Recorded data from the SMILE sensor

For the LCF method we need to get a stair shaped plots to ensure that we can use LCF to find the calibration matrices.

The measured and real data are fitted on the regression line. If the measurements are taken while rotating then the method will not work. The algorithm can be used on both magnetometers. To demonstrate how it can be implemented we can use the measurements from test 3 shown in Fig. 16. We take the measurements from each axis when the magnetic field is changing in that axis. We plot it alone and then we take the mean values of each step highlighted in red in Fig. 8. These mean values are be our expected values μ_0 . The changing magnetic field from $-60\mu\text{T}$ to $60\mu\text{T}$ in 16 steps that we applied using Helmholtz coils are our real values. We use the inbuilt command polyfit in Matlab that performs the linear regression. To make sure an accurate calibration was performed using the LCF, we plot the output from playfit command with the real values using polyval and if the data points line up with the regression line or close to it as shown in Fig. 9 then it will provide good calibration.

For the ALS Implementation, the algorithm was provided by former members of the ADCS (Attitude determination and control systems) team in MIST project. A section on how the algorithm can be used to write a code exists in [8].

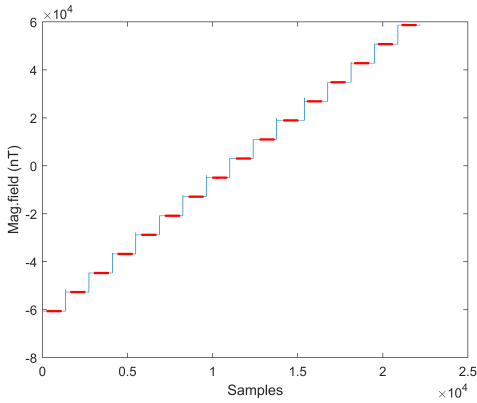


Fig. 8: The red data line are used to find the mean value of each step.

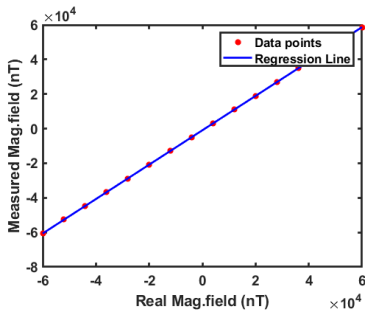


Fig. 9: An example of a good linear regression as all the data points lined up on the regression line

IV. RESULTS

A. Testing software

The first test is one point measurement in the Lab without moving the magnetometer. The result from the first test is

shown in Fig. 11. Both calibrated and uncalibrated measurements align on the same point on comparison to the sphere and concentrated in one point. The results of test 1 was used to confirm that the software is working well and no errors or major issues can be found.

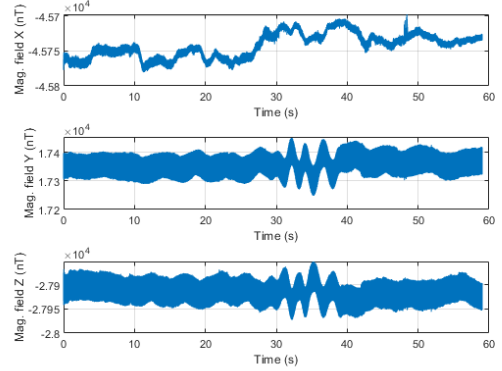


Fig. 10: Uncalibrated data from test 1 SMILE

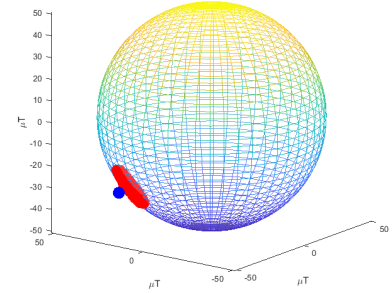


Fig. 11: Test 1 Ellipsoid fitting using SMILE

B. Graphical and numerical results of the calibrations

In Fig. 13 and Fig. 14, the graphical results ellipsoid fitting calibration for test 5 and 6 are presented. The blue dots represents the uncalibrated data and the red dots are the calibrated data. Some clear differences between the two results can be noticed such as the results from test 5 where the distance between calibrated and uncalibrated data is short. While the results from test 6 are clearer and have big differences between the uncalibrated and calibrated data. The data points in test 6 forms dots instead of a continuous line due to low sampling rate in the magnetometer in IMTQ. The data from test 2 is less than test 5 due to shorter length of measurement time. In Table II the numerical outputs of the ALS and LCF algorithm are presented. The s_x , s_y and s_z are the scaling factors and they are the diagonal elements in (1) and they are constants. The off_x , off_y and off_z are the offsets in each axis from (2).

Table II
Numerical results with scaling factors (constants) and offsets in μT

Test number	Magnetometer	Method	s_x	s_y	s_z	off_x	off_y	off_z
2	SMILE	ALS	0.91048	0.935214	0.910972	-1.86214	-1.3043	-4.01389
3	SMILE	LCF	0.9906	1.0038	0.9942	-1.1237	-0.9107	-0.8870
4	IMTQ	LCF	1.3358	1.3907	1.0883	12.8052	19.5783	0.9405
5	SMILE	ALS	0.9909	1.0043	0.9950	-1.12865	-1.0229	-1.03324
6	IMTQ	ALS	1.2071	1.2495	1.1409	13.3243	19.1920	0.36187

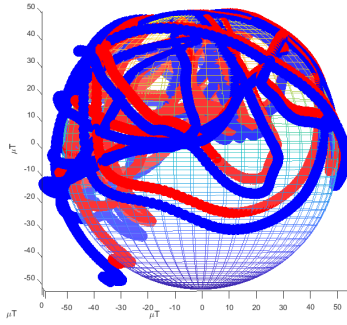


Fig. 12: Results from Test 2

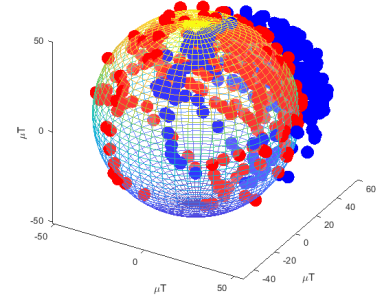


Fig. 14: Results from Test 6

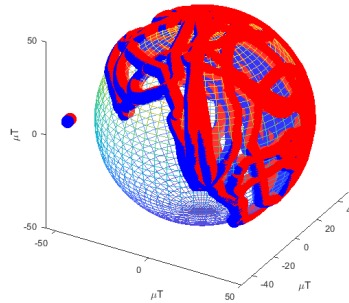


Fig. 13: Results from Test 5

V. DISCUSSION

A. Difficulties and limitations during tests

The tests were hard to perform as they required a lot of equipment. Other limitations that affected the measurements were the safety measures for one of the magnetometers. The IMTQ magnetometer will be used in the MIST satellite and the risks of damage or static electric charges is high and had to be avoided as much as possible. The SMILE sensor had some difficulties in the beginning as it also required some equipment to start up and they were not available.

B. Analysis of the measurements

As we can see in Fig.10 and Fig.11, we have the behavior that was expected from the SMILE magnetometer. Test one was mostly done to confirm the integrity of the software. The first test as shown in Fig.11 has the measurements concentrated in one point. The visualisation of the measurements also matches

the the movement of the magnetometer as it did not move and took measurements in one point. Thus, the visualisation of the measurements is accurate. The second test was done in the same place as the first test but while rotating the magnetometer and the visualisation provided by calibration algorithm gave an accurate representation of the rotation. In Fig.10 we see the measured magnetic field plotted on the time vector. We notice the plots in Fig.10 are thicker than the plots in Fig.16. This is due to the fact that the measurements are less spread out and the duration of test 1 was longer. One similarity we can see between test two and five, is that the differences between the calibrated and the uncalibrated measurements are not that big compared to the results from test 6. One possible explanation is that the SMILE magnetometer is a scientific magnetometer used to measure magnetic field accurately with details while the IMTQ has an inbuilt attitude determination magnetometer which is mass produced and does not need as much detailed measurements as the SMILE sensor. To compare between the two magnetometers, we can look at Fig.13 and Fig.14, as test 5 and 6 have been conducted in a similar way and the same place. One big difference is the form of the data in the figures. We see that Fig.13 which is data recorded by SMILE has the measurements form a line while in Fig.14 we have data in the form of points. That is due to the SMILE sensor having high sample rate of 250 sample per second compared to the IMTQ sample rate which is 2.5 per second. For the Helmholtz coils tests using, we see a clear patterns in both tests. Both Fig.15 and Fig.16 gives a good representation of the applied magnetic fields that we tested. The IMTQ was not aligned with the axes of the Helmholtz coils which made the applied magnetic field from x and y axes affect each other.

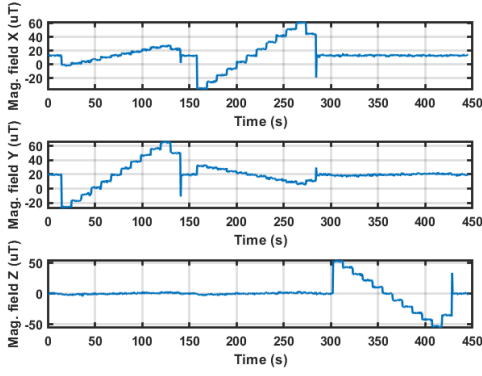


Fig. 15: Helmholtz coils Measurements with IMTQ Test 4 ($uT = \mu T$)

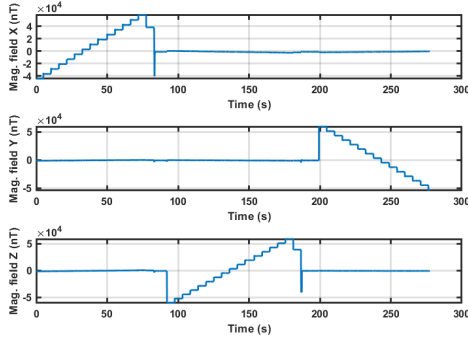


Fig. 16: Helmholtz coils Measurements with SMILE Test 3

C. Analysis of the results and comparisons

For the numerical results shown in Table II, we can see that the calibration matrices are close to each other. The SMILE calibration matrices from test 3 and test 5 in Helmholtz coils are really close to each other, especially the scaling factors with highest difference in s_z being 0.0008. For the calibration matrices from test 4 and 6 using IMTQ, we can see that the differences are also small but not as small as the ones shown from test 4 and 5. That is due to the IMTQ having high biases compared to the SMILE sensor. The only clear difference is between the calibration matrices in MIST lab and Helmholtz coils. That was expected due to the lab having a lot of disturbances and electric circuits and current going around in the lab while inside the Helmholtz coils the external disturbances are canceled the Helmholtz coils itself. Another interesting thing to notice is the offsets from IMTQ are really high in both methods which confirms the IMTQ has a really high biases in the magnetometer and that is not the result of errors in the methods used. Especially off_y which is the offset in y axis measurements being as high as $19 \mu T$. To really be able to test and see how good do these methods compare to each other. We use the rotation measurements from test 5 and 6 and calibrate them using ALS and then we calibrate them using the calibration matrices we got from LCF (step) again by inserting them in the mathematical model (4) with the measurements. Since we know what the applied magnetic field in these tests which was $50 \mu T$ applied by Helmholtz

coils, it is expected that when we calibrate them and plot the magnitude of the calibrated data, it should converge to $50 \mu T$. That is shown in Fig.17 where we can see that both methods have converged to $50 \mu T$. The ALS seems to be a lot more accurate as it is converging more specifically to $50 \mu T$ while the LCF (step) is changing between $49 \mu T$ and $51 \mu T$. When compared to the raw data, we see that both methods got closer to the real magnitude which is an indicator that both methods are good and have corrected biases in the magnetometers.

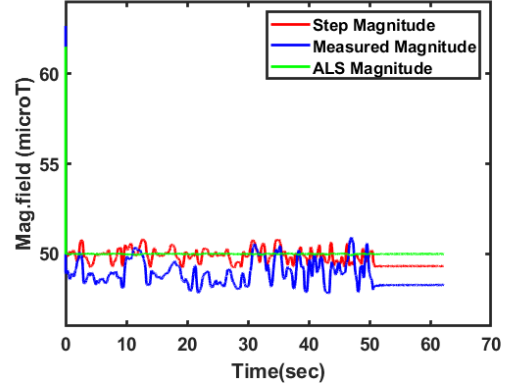


Fig. 17: Results comparison for the calibration using SMILE

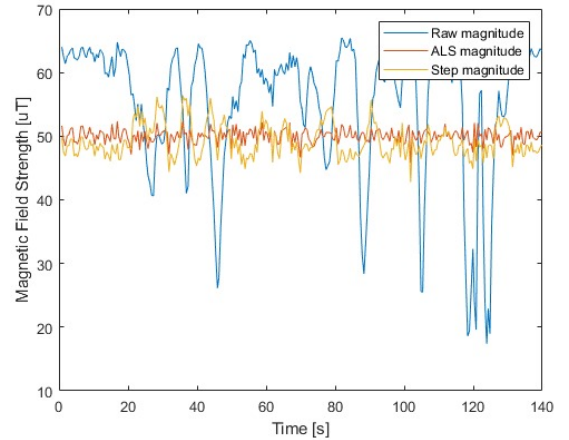


Fig. 18: Results comparison for the calibration using IMTQ

We can see in Fig.18 that the IMTQ has really high biases based on how the raw data is changing a lot and not close at all to the real data compared to SMILE which had raw data changing between $47 \mu T$ and $50 \mu T$. In 17 we see the magnitude of the magnetic field in the first second is as high as almost $70 \mu T$ for ALS and raw data. This is just false measurement from the magnetometer itself as we checked the raw measurement data and it was measuring $70 \mu T$ for an unknown reason. For the comparison between the methods we see the same conclusion as the one we got from the SMILE comparison. They are both good and got closer to the real data. However, ALS is converging more precisely to $50 \mu T$ which means that it is a more accurate method than LCF using Helmholtz coils.

D. Future Work

The algorithms provided good outputs and results but i think it could be developed further. For example the ellipsoid fitting algorithm has a main requirement and that is the magnetic field has to be constant. The algorithm could be possibly further developed by taking every measurement and comparing it to more than one ideal sphere, which can help with calibration in places where the magnetic field is changing. The other possible development is looking for accurate way to determine the offsets by the hard and soft iron in the satellite. One possible method could be by using Solid edge which is a cad program to simulate electric current in the satellite. This way we could find the more small offsets that the satellite will produce and take it into account. The errors from the Helmholtz coils when applying the magnetic field should be included in the algorithm for LCF (Step) when calibrating so we can get more accurate offsets .

VI. CONCLUSION

The Helmholtz coils calibration is an effective and useful way to calibrate the magnetometer in the satellite as a whole without the need to rotate it and risk damages. The ellipsoid fitting method is also a useful and more accurate method that is cheap to perform without the need of big and expensive equipment like Helmholtz coils. Since the satellite is not yet integrated and the mechanical simulations are not done, assessing the risks of rotating it is hard to do. In the end, we can say that the process of rotating the satellite can be risky taking into consideration its high financial costs. The goal of this project was to test LCF using Helmholtz coils method and compare it to the ALS with ellipsoid fitting, and we can see that we managed to do it and record the procedure as a test calibration to be used in the calibration of the magnetometer in satellite as a whole in December 2022.

ACKNOWLEDGMENT

I would like to thanks my advisors Nicolay Ivchenko and Sven Grahn for their continuous support through out the project.

REFERENCES

- [1] . (2022, Feb.) Basic facts. Stockholm, Sweden. . [Online]. Available: <https://mistsatellite.space/basic-facts/>
- [2] M. Kok and T. B. Schön, "Magnetometer calibration using inertial sensors," *IEEE Sensors Journal*, vol. 16, no. 14, pp. 5679–5689, 2016.
- [3] E. Bronaugh, "Helmholtz coils for calibration of probes and sensors: limits of magnetic field accuracy and uniformity," in *Proceedings of International Symposium on Electromagnetic Compatibility*, 1995, pp. 72–76.
- [4] —. (2022) What is calibration? Norwood, MA, United states of America. . [Online]. Available: <https://www.aicompanies.com/education-training/calibration/what-factors-affect-calibration/>
- [5] Forslund, S. Belyayev, N. Ivchenko, G. Olsson, T. Edberg, and A. Marusenkov, "Miniaturized digital fluxgate magnetometer for small spacecraft applications," *Measurement Science and Technology*, vol. 19, p. 015202, 12 2007.
- [6] Vidya Prabhu. (2021, 16) What is a magnetometer and how does it work? 6920 Koll Center Parkway, Suite 219 Pleasanton CA 94566 USA. . [Online]. Available: <https://www.youngwonks.com/blog/What-is-a-Magnetometer-and-How-Does-It-Work#:~:text=Magnetometers%20using%20the%20Hall%20effect%3A&text=To%20put%20it%20simply%2C%20the,field%20perpendicular%20to%20the%20current./>
- [7] Q. Gao, D. Cheng, Y. Wang, S. Li, M. Wang, L. Yue, and J. Zhao, "Compensation method for diurnal variation in three-component magnetic survey," *Applied Sciences*, vol. 10, no. 3, 2020. [Online]. Available: <https://www.mdpi.com/2076-3417/10/3/986>
- [8] I. Markovsky, A. Kukush, and S. Huffel, "Consistent least squares fitting of ellipsoids," *Numerische Mathematik*, vol. 98, 07 2004.
- [9] Z. S. Kukush, A., "Kukush,a., zwanzig, s.: On consistent estimators in nonlinear functional eivmodels." In: *S. Van Huffel and P. Lemmerling, (eds.), Total least squares and errors-in-variables modeling: Analysis, Algorithms and Applications*, pp. 145–155, 2002.
- [10] P. Mahavarkar, J. John, V. Dhapre, V. Dongre, and S. Labde, "Tri axial square helmholtz coil system at the alibag magnetic observatory: Upgraded to magnetic sensor calibration facility," *Geoscientific Instrumentation, Methods and Data Systems Discussions*, pp. 1–11, 11 2017.
- [11] E. G. Blom, G. Enger, J. and Holst, L., "Regression analys," in *San-nolikhetsteori och statistikteori med tillämpningar*. Lund, Sweden: Studentlitteratur AB, 2017, pp. 359–360.
- [12] G. Hunter, "Linear Curve Fitting," 06 2018. [Online]. Available: <https://blog.mbedded.ninja/mathematics/curve-fitting/linear-curve-fitting/>
- [13] M. Saqib, F. S. N., and F. J. N., "Design and development of helmholtz coils for magnetic field," in *2020 International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE)*, 2020, pp. 1–5.

Wave Propagation Experiment on FPGA with Miniaturized Payload for Sounding Rocket Application

Leonardo Filippeschi

Abstract—This bachelor’s thesis aims to implement a wave propagation experiment on Field-Programmable Gate Array to detect the signal strength at pre-defined frequencies for use in sounding rocket experiments. This includes the choice of suitable components such as analog to digital converters, filters, voltage regulators, and amplifiers. The board prototype was designed by keeping in mind the need for a miniaturized solution that would still provide the wanted results, by following design guidelines. The second part of the project involves the design of the software in a hardware description language. An analysis in MATLAB® was done to determine the parameters needed to successfully reconstruct the transmitted signal on the receiver, while still being able to fit on the given FPGA. To make sure of that, a simulation was performed on ModelSim a tool for simulation and debugging for VHDL. From the simulations, it can be concluded that this design is feasible and that this project gives the basis for further development, to create a viable solution for a wave propagation experiment with a miniaturized payload.

Sammanfattning—Denna kandidatuppsats syftar till att implementera ett vågutbredningsexperiment på Field-Programmable Gate Array för att detektera signalstyrkan vid fördefinierade frekvenser för användning i sonderingsraketexperiment. Detta inkluderar val av lämpliga komponenter som analog till digital omvandlare, filter, spänningsregulatorer och förstärkare. Kortprototypen designades genom att ha i åtanke behovet av en miniaturiserad lösning som fortfarande skulle ge önskat resultat, genom att följa designriktlinjerna. Den andra delen av projektet involverar design av programvaran i ett hårdvarubeskrivningsspråk. En analys i MATLAB® gjordes för att bestämma parametrarna som behövs för att framgångsrikt rekonstruera den sända signalen på mottagaren, samtidigt som den fortfarande kan passa på den givna FPGA. För att säkerställa det gjordes en simulering på ModelSim ett verktyg för simulering och felsökning för VHDL. Från simuleringarna kan man dra slutsatsen att denna design är genomförbar och att detta projekt ger grunden för vidareutveckling, för att skapa en hållbar lösning för ett vågutbredningsexperiment med en miniaturiserad nyttolast.

Index Terms—Wave propagation experiment, REXUS/BEXUS, FPGA, IQ demodulation, ADC, VHDL.

Supervisor: Nikolay Ivchenko

TRITA number: TRITA-EECS-EX-2022:158

I. INTRODUCTION

As the number of satellite launches and flights increases each year, it is becoming more and more important to understand how space weather affects radio communication, navigation, and other areas of human activity. The main source of the variation in space weather is caused by solar

UV radiation and energetic auroral particles which ionize the molecules and atoms in the upper atmosphere, maintaining the layer of free charge carriers, also known as the ionosphere. Understanding and measuring the electron concentration of the ionosphere is fundamental to us as it plays an important role in telecommunications, and space weather and it constitutes the boundary between the vacuum of space and the lower atmosphere, where we live and breathe. It is important, as an example, in radio communication, where High Frequency (HF, or shortwave) radio waves are refracted and reflected, due to the presence of ionized particles in the transmission path. From this fact, the properties of plasma on the path can be derived. This has been explored in various ways. An approach, which has been successfully used on sounding rockets and which will be explored in this thesis project, is to use a ground-based HF transmitter of linearly polarized electromagnetic waves and detect the signal strength at pre-defined frequencies of the wave onboard the rocket. From these observations, the altitude profile of the electron concentration can then be reconstructed.

The division of Space and Plasma Physics at KTH Royal Institute of Technology has a long history of experiments launched aboard a sounding rocket in the REXUS/BEXUS programme, which is realized under a bilateral agreement between the German Aerospace Center (DLR) and the Swedish National Space Agency (SNSA), giving the opportunity to elected teams to launch their experiments from the Esrange Space Center in northern Sweden twice a year [1]. This project aims to be implemented in the form factor of the PRIME experiment, flown in 2019, whose aim was to measure plasma parameters in the lower ionosphere with the use of Langmuir probes [2] in the form factor of a miniature Free-Falling Unit (FFU).

The purpose of this form factor was to validate the feasibility of an experiment that would comply with the required dimensions of the DART rocket. DART is a small launch vehicle developed by T-Minus Engineering, whose aim is to provide an affordable alternative for delivering small payloads to altitudes above 120km [3]. This would allow for more frequent and more affordable measurements of the upper atmosphere.

This thesis project aims at building upon previous experiments and designing a viable solution for a wave propagation experiment, which can then be implemented in future projects.

Abbreviations

ADC	Analog to Digital Converter
FPGA	Field Programmable Gate Array
FFU	Free Falling Unit
HDL	Hardware Description Language
HF	High Frequency
IC	Integrated Circuit
LDO	Low Dropout Regulator
MSPS	Mega Samples Per Second
μC	Micro Controller
PCB	Printed Circuit Board
REXUS	Rocket EXperiment for University Students
Radio Frequency	RF
Signal to Noise Ratio	SNR

II. BACKGROUND

A. Ionosphere

According to the IEEE Std 211-2018, the Ionosphere is defined as that part of a planetary atmosphere where ions and free electrons are present in enough quantities so that they are able to affect the propagation of radio waves [4]. Due to its varying nature, it is not always possible to make a clear distinction between where the ionosphere starts and ends. It is however been approximated to be roughly bounded between 50 km and 1000 km above the ground level, where the ionosphere transitions to the plasmasphere [5]. Good knowledge of the electron concentration is key to the creation of reliable theoretical models. Its understanding is also important for satellite navigation, where the delay caused by the ionosphere makes up for the main uncertainty [6].

With the advent of sounding rockets, it has become possible to study parts of the mesosphere known as the D-region, which is the first layer where free electrons appear, and the E-region. To study the concentration of free particles in situ, multiple types of probes have been adopted and successfully tested. These include Langmuir Probes, capacitance probes, positive ions probes, and the wave propagation experiment. The latter, which is the one being investigated in this thesis project, consists of transmitting a linearly polarized signal from a ground transmitter and then receiving the signal aboard the rocket. Due to the interaction with the ions and free electrons on its path, the signal is affected, and by measuring its strength and polarization, the electron content between the ground and the payload can be reconstructed. Various frequencies have been used to obtain an optimal coverage for different altitudes, the reason for this is that depending on the angle of attack and frequency, the signals are either reflected or pass through the electron concentration without being affected. Tests have been performed with frequencies of 1.3 MHz, 2.2 MHz, 3.883 MHz and 7.835 MHz. Higher frequencies have also been tested but the results did not show any major improvements when compared to the 7.835 MHz frequency [6].

B. Student Workshop

To solve the task of the project, it is possible to use the Student Workshop at KTH [7], where various tools, materials, and machines are available to use by students who are involved

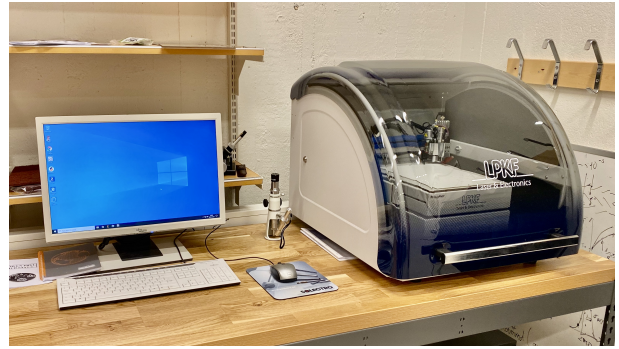


Fig. 1. PCB mill available in the student workshop, taken from [8].

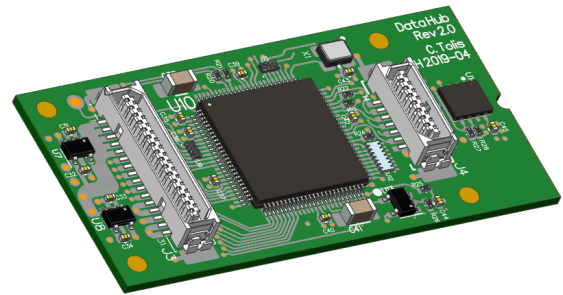


Fig. 2. Render of the Data Hub from the PRIME project [9].

in projects. Some of the tools include oscilloscopes, power supplies, soldering stations, 3D printers, PCB mill, and more. The availability of the PCB mill, shown in figure 1, makes it possible for us to manufacture a prototype in-house. This is a great opportunity, which not only reduces costs but also allows us to be able to reiterate the design faster if the need arises. The only problem with such a solution is that vias are only possible to be done manually and only double layer designs are allowed which makes it more challenging in some parts of the design.

III. REQUIREMENTS

In this section, the given requirements that this project had to respect are outlined. These included the use of previously validated electronics, the sampling rate of the receiver, and the form factor of the PRIME experiment, which this project aimed to be implemented in.

A. Previous electronics

Legacy hardware and designs from previous experiments were available for the use of this project, which could be analyzed to draw inspiration from and reused where applicable. The main part of the electronics system that was re-used is the Data Hub, designed by Christos Tolis for the PRIME experiment [9] shown in figure 2. The Data Hub is where the FPGA, microcontroller (μC), and SD card are mounted together with other sensors. It was responsible for the data saving and the operation of the experiment. It also gave the main share of constraints to this project as the size of the FPGA is decided by

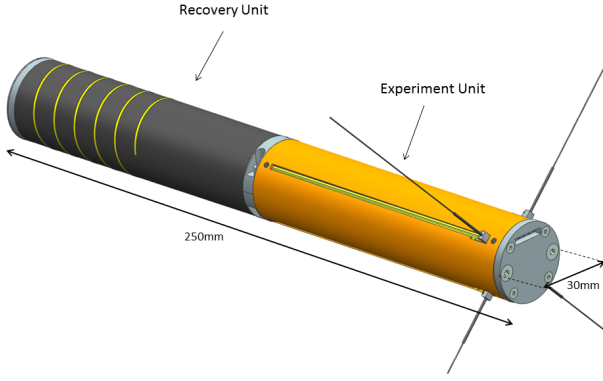


Fig. 3. Render of the Free Falling Unit from the PRIME project [12].

the one used in the Data Hub, namely the A3P250-VQG100I from Microsemi [10]. This model offers 250,000 system gates, equivalent to roughly 3k logic elements and 36 kbits (1,024 bits) of RAM. The maximum allowed speed is 350 MHz. No Digital Signal Processors (DSP) are offered on this model and this makes it even more important to keep the design as small as possible. DSPs are specialized blocks on the FPGA that are used for operations such as multiplications, divisions, additions, and more. The fact that they are optimized for these operations makes it possible to reduce the space occupied when compared to similar operations without the use of DSPs. Another constraint is given by the available pins provided by the Data Hub, only 27 of the total pins are re-programmable and can be used for either Input or Output.

B. Sampling rate

The sampling rate was the most important requirement for this project as it allows for the correct acquisition of the signal. The Nyquist–Shannon theorem states that an input signal can be reconstructed with no loss of information as long as it is sampled at a frequency greater than or equal to twice the original frequency [11]. Since the transmitted frequencies do not go higher than 7.835 MHz a sampling rate of at least 15 MSPS was needed.

C. Form factor

As mentioned in section I, the aim of this project is also to create a board that would ultimately fit in the PRIME Free Falling Unit (FFU) shown in figure 3. The form factor is quite restrictive and makes it important to choose adequate components that would fit. Ideally, after the prototype board has been validated, the design should fit in a form factor similar to the one of the Data Hub which would stack on top of it. The use of a multi-layer PCB is a must in this case.

IV. POSSIBLE SOLUTIONS

To implement the wave propagation experiment, three solutions were considered to accomplish the task. These included a faster data saving solution, while the other two are based on the detection of a chosen frequency with its relative filtering

and detection done either in an analog or in a digital way. Due to my background and interest, the analog path was discarded. After extensive initial research on the super-fast saving solution, this path was also discarded due to the impracticality of the current form factor. However, it gave some inspiration for possible future experiments with the use of an SD Express card [13]. This option was explored because of its form factor, which is the same as that of a regular SD card, already being used on the current revision of the Data Hub. The speeds however are two orders of magnitude higher going from an advertised speed of 12.5 MB/sec to speeds ranging from 985 MB/sec to 3940 MB/sec depending on the type and amount of PCIe lanes being used. The increase in speed is given by the use of the PCIe standard when compared to the SD standard, this would be a viable solution but with the size and architecture of the current FPGA and μC , it would have not been possible to implement in the time of this project. Furthermore being a new technology, its availability was not guaranteed and it could have been hard to test it. This solution would have required a full redesign of the current system and it would have gone outside of the scope of a bachelor's thesis, for these reasons this option was put aside. The digital solution was therefore chosen.

A. Digital solution

The digital solution consists of filtering and amplifying the acquired analog signal. Once this has been done a suitable ADC has to be used to convert the analog signal to digital which is then read by the FPGA and manipulated to extract the amplitude of the chosen frequency. The acquisition steps are reported in figure 4, where the signal goes first through a pre-amplifier and then for the second stage, through a differential amplifier and low pass filter at 15 MHz. The expected voltage from the antenna is in the range of μV , as reported in [14], while the ADC has an input of $\pm 1V$. These parameters lead to an approximate gain of 10^6 , for this reason, a double stage amplifier was chosen, to be able to attain a higher gain than a single amplifier without having to push it to its limits. If a high gain is desired for high bandwidth signals, the choice of a multi-stage amplifier becomes even more important. This is because bandwidth and gain are inversely proportional to each other and one cannot be achieved without affecting the other parameter.

Once the signal has been sampled it is time for the FPGA processing chain to extract the signal strength at pre-defined frequencies and save the result. To do this an IQ demodulation has been chosen as a solution. The overall structure of such an algorithm is reported in figure 5.

The signal to be reconstructed is in the form:

$$A \cos(2\pi ft + \phi), \quad (1)$$

with f being the chosen frequency, which can also be rewritten as

$$A \cos(2\pi ft) \cos(\phi) - A \sin(2\pi ft) \sin(\phi), \quad (2)$$

where the In-Phase (I) signal can be assigned to be:

$$I = A \cos(\phi), \quad (3)$$

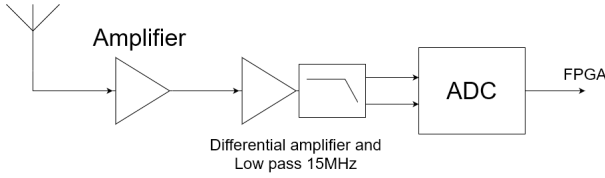


Fig. 4. Acquisition of signal chain.

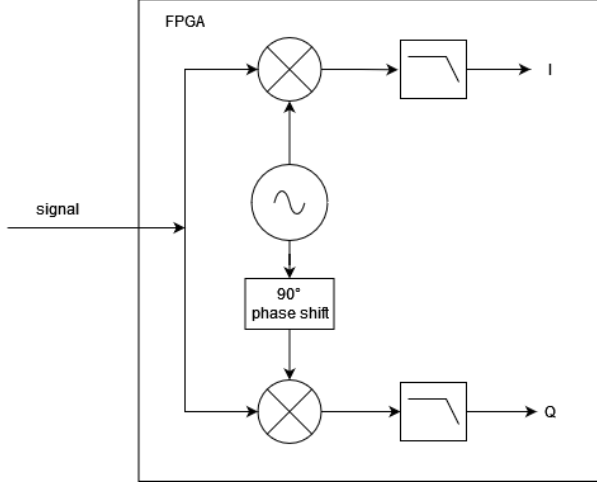


Fig. 5. IQ demodulation chain for acquired signal.

and the Quadrature (Q) signal as:

$$Q = A \sin(\phi). \quad (4)$$

Giving the signal in the final form:

$$I \cos(2\pi ft) - Q \sin(2\pi ft). \quad (5)$$

If now the input signal is multiplied by the signal generated on the FPGA with the same chosen frequency the results is:

$$\begin{aligned} & A \cos(2\pi ft) \cos(\phi) \times \cos(2\pi ft) \\ &= (I \cos(2\pi ft) - Q \sin(2\pi ft)) \times \cos(2\pi ft) \\ &= I \cos(2\pi ft) \times \cos(2\pi ft) - Q \sin(2\pi ft) \times \cos(2\pi ft) \\ &= I \times \frac{1}{2} \times (\cos(0) + \cos(4\pi ft)) \\ &\quad - Q \times \frac{1}{2} \times (\sin(4\pi ft) + \sin(0)) \\ &= I \times \frac{1}{2} \times (1 + \cos(4\pi ft)) - Q \times \frac{1}{2} \times (\sin(4\pi ft) + 0) \\ &= \frac{1}{2} \times (I + I \times \cos(4\pi ft) - Q \times \sin(4\pi ft)) \end{aligned} \quad (6)$$

By then applying a Low Pass Filter on the output, the components with a high frequency can be removed, leaving just the I signal. The same operation can be done with the phase-shifted signal, which then results in the Q signal as the output. In the chosen design the low pass filter was implemented as an accumulator over multiple periods, effectively removing the high frequency components.

V. IMPLEMENTATION

A. ADC

When choosing the ADC, some aspects were most important, the format of the output, how well it fits the chosen setup, and how fast it could sample the analog signal. When looking at the output of an ADC, there are multiple options, it can either be parallel, serial or a mix of the two. In this case, since an FPGA is used where multiple pins can be configured as an input and due to the fast nature of the processing required, a fully parallel type was the most suitable. As previously stated in subsection III-B, the minimum sample rate had to be of at least 15 MSPS. When sampling fast signals, it is recommended to use differential inputs. This is done to make it more immune to noise since noise is present on both signals, and when in differential mode only the difference between the two leads gets measured, resulting in the subtraction of the noise and therefore better Signal to Noise Ratio (SNR). When it comes to the choice of type of ADC, various options were available: Successive Approximation (SAR), Delta-sigma, Dual Slope, Pipelined, and Flash. After extensive research, the LTC1744 from Linear Technology [15] was chosen. It is a pipelined ADC with 14 bits of precision at a sample rate of 25 MSPS, which provides great flexibility in terms of input and output. The input is differential and selectable at the values of $\pm 1V$ and $\pm 1.6V$. The output voltage is also selectable between the values of 0.5 V and 5 V with a fully parallel configuration, an overflow flag is available together with a clock out signal that indicates when the sample is ready to be read. The sampling rate is selectable through the use of the ENCODE signal which can go down to 1 MSPS, providing great flexibility.

B. Filter

Filters are necessary both at the input of the signal and for the IQ demodulation. At the input, the low pass filter is used to reduce all the possible noise and high frequencies that could be induced and picked up. The one chosen for this project is the LT6600 from Linear Technologies, which is a differential amplifier and 15MHz Lowpass Filter [16]. In the IQ chain, the filtering is done digitally on the FPGA as explained in IV-A.

C. Amplifier

The chosen amplifier is the OPA858-Q1 from Texas Instruments [17]. This operational amplifier was chosen for its high Gain Bandwidth Product, low input voltage noise, and high slew rate. All these parameters are important for getting a good signal that is not affected by the noise and that is fast enough to follow the input signal. The main reason for the choice of this amplifier was however its gain response. It can hold high gain values even for high frequencies, which others struggle with. The difficulty with choosing this component is that it comes in a WSON package, which is not the easiest to hand solder on a prototype board.

D. PCB design

When designing a PCB there are a few guidelines that should be followed to prevent noise and Electromagnetic Interference (EMI) between components, especially when dealing with high-speed signals. One of the most important is that signal current loop areas should be minimized. Traces and components should be as straight and as short as possible. The reason for this is to prevent coupling with other circuits and avoid possible radiations. Traces should never run across split planes unless a stitching capacitor is used [18], which helps with the return path.

Another important aspect to keep in mind is that vias and components should always be placed as close to the pins as possible. Analog and digital planes should be kept separate when possible to avoid interference. Bypass capacitors had to be put at the inputs of the voltage supply of the ADC and other components. This serves two purposes. The first purpose is to smooth out possible voltage drops, the reason for this is that capacitors placed close to the pins can supply extra current when it is needed, without having to fetch this current from the source, resulting therefore in smoother operation and better response. The second purpose is to partly filter out ripple frequencies on the power lines, depending on the size of the capacitor, either low or high frequencies will be attenuated, resulting in a cleaner voltage for driving the devices. As an example for 1 MHz bandwidths a $1\ \mu\text{F}$ is used while for 10 MHz a $0.1\ \mu\text{F}$ is preferred. It is also usually better to use multiple smaller values capacitors rather than a bigger one with the same capacitance [19].

An array of resistors is also recommended to be used at the output of the ADC. This is done for two reasons. The first one is to reduce the ringing noise on the line. Ringing noise appears when a signal bounces back and forth on the line, causing possible problems such as the wrong detection of highs and lows and unstable outputs. By applying a resistor in series on the line, this effect is reduced at each bounce until it dies out. The second minor reason can be to lower the current that flows towards the input.

When using an antenna connector, special care must be put into the calculation of the matched impedance. This is done to respect the maximum power transfer theorem, which states that the maximum power is transferred from the source to the load if the internal and load resistances are matched. To do this, there are online calculators that use Wadell's equations, which were published in the Transmission Line Design Handbook [20].

All these precautions were applied when designing the prototype board using Mentor Graphics® Xpedition from Siemens, the result is shown in subsection VI-B and the schematics can be found in appendix A.

E. Software

Software for the FPGA was written in VHDL using Libero® SoC Design Suite from Microchip [21]. The high-level flow chart is represented in figure 5 and the software has been written according to that. When dealing with FPGA a few precautions had to be taken into consideration. These included

two's complement representation and possible ways to optimize the structure, with tricks that could be done to make the implementation faster at the expense of occupying more space. In a two's complement representation the most significant bit (MSB) represents the maximum negative number, the rest of the bits are then added following a regular unsigned number representation. For this reason, special care had to be taken when dealing with operations not to overflow into the most significant bit. The same applied when adding a positive and a negative number of different sizes. The negative number had to be padded with 1's compared to 0's when dealing with an unsigned number.

Further precautions have to be considered when dealing with hardware that is intended to be used in environments with higher amounts of radiation such as space. Not only the type of FPGA is important but also special software design techniques are important for the system to be immune from radiation. These aspects are explained in great details in [22] and should be taken into consideration for the final design.

Another aspect that had to be taken into consideration when dealing with binary operations is the problem of overflows and the variable amount of bits required for different operations. Given the constraints of the FPGA, care should be put into choosing the correct amount of bits to make sure that the design would fit on the chosen FPGA and that data saving would be fast enough. In the chosen design, the operations that required the most space are multiplication and an accumulator. As an example when multiplying a 14 bits number with a 10 bits one, yields a 24 bits result.

More important choices were around the size of the cosine and sine signals generated by the FPGA to be multiplied with the received signal and how many periods to integrate over to be able to save data less frequently, due to the limitations on the data saving. These were key because they define how good the quality of the received signal is.

In order to have more qualitative measures, an analysis was performed. The parameters investigated are the number of bits used for the generation of the signals and the number of periods used for the integration. The measurements are based on two key parameters, namely the δF and the δA shown in figure 6. The δF measures the difference in the picked-up frequency, the narrower, the better, and it is measured at half of the peak amplitude. The δA measures the maximum amplitude of the reconstructed signal at a distance greater than $\frac{f}{2}$ from the chosen frequency f_0 . Points have been calculated for different numbers of periods and then a spline interpolation has been used to connect the points. From figure 7, it can be seen how the number of bits doesn't affect the passband frequency but only the number of periods. On the other hand in figure 8, the effect of the number of bits is clear as it helps in obtaining the correct amplitude of the signal. Having a smaller amplitude for the δA , results in less noise on the output, giving a cleaner signal.

To illustrate the functionality, a simulation has been done in MATLAB® to see how well different configurations perform. A signal was generated including different Gaussian envelopes with two components at the correct frequency and one at the wrong frequency. A two's complement representation of the

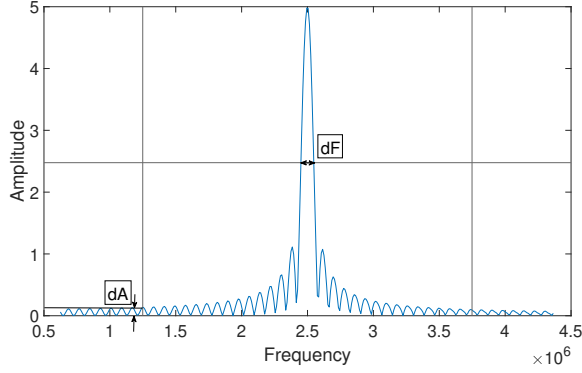


Fig. 6. Explanation of the values δF and δA used for the qualitative analysis performed in MATLAB®.

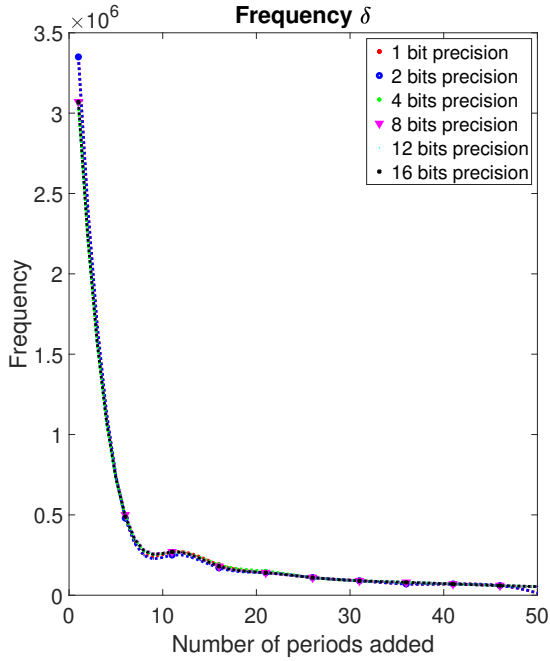


Fig. 7. Results from the analysis done in MATLAB® of the δF parameter.

sine and cosine signals was then created and used to evaluate the results. Different parameters were investigated. In figure 9, the generated signal can be seen in the top panel, below that from left to right the reconstructed I, Q components and the amplitude can be seen. From the illustration, the results match the results obtained from the analysis. One period is not enough to remove the signal with the wrong frequency and more bits are needed to maintain the correct amplitude.

For the final design, the choice to use 4 bits values and integration over 25 periods was done, as it strikes a balance between achieving enough precision in the reconstruction of the amplitude and in not creating values that would be too large.

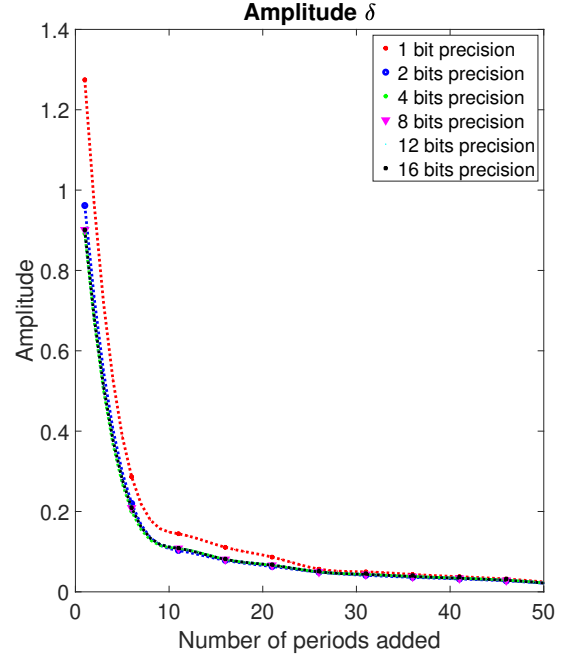


Fig. 8. Results from the analysis done in MATLAB® of the δA parameter.

F. VHDL

To double-check the design, a testbench has been used throughout the development to spot mistakes and validate the implementation. A testbench is a simulation, where made-up values, also known as stimuli, are generated and fed into the algorithm. The results are then stored and displayed as if they were run in real-time. To create the input signal a script was written in MATLAB® to auto-generate the VHDL code. The generated stimulus was made to resemble the output of the ADC, using a 14bits value in two's complement format. The advantage of this implementation is that it gives the flexibility to rapidly define new arbitrary signals and test them without having to generate these signals in real life which would be much more complicated.

VI. TESTS AND RESULTS

A. Testbench

The results from the simulation are presented in this section. In figure 11 the generated signals and output of the multiplications are shown. The generated signals for the sine and cosine signals are shown in figure 10. The VHDL implementation has been done using the SmartDesign provided in the Libero IDE and it is shown in appendix C. It is useful to look at that design as successive stages, where the calculations are performed before each new rising edge of the clock so that the propagation delays have time to settle. Each new result is latched at the next clock. This design results in a delay of the result at the output but it doesn't affect it. The first stage was the acquisition of the signal from the ADC and the generation of the sine and cosine wave samples by using lookup tables as inspired by [22]. In the next stage, the multiplication between these two values was performed. After the multiplication has

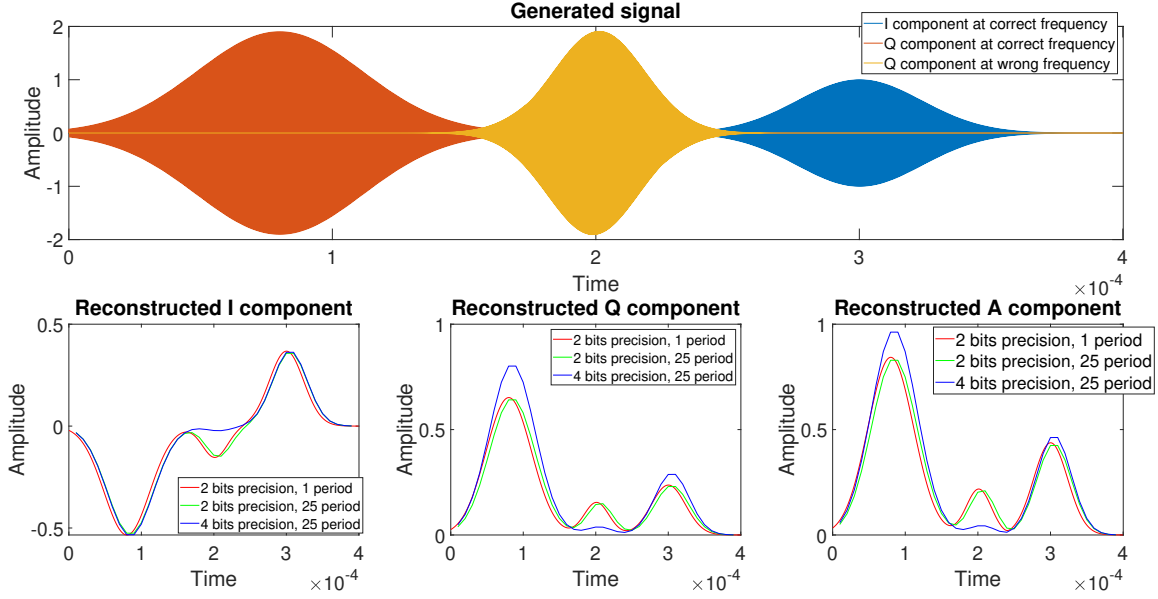


Fig. 9. Illustration done in MATLAB[®] of transmitted and acquired signals. In the top panel the generated signal is shown with 3 envelopes, two of which are at the correct frequency. In the bottom part the reconstructed I, Q and amplitude are shown from left to right. Different parameters were used for the investigation, which are reported in the bottom figures.

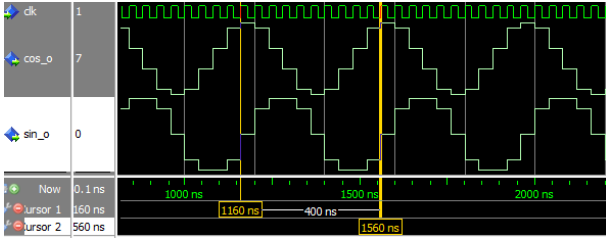


Fig. 10. Cosine and sine wave signals generated by the FPGA with a frequency of 2.5 MHz.

been done and saved successfully at the next clock, the result was added to the sum. For the final stage, the number of periods was counted and when the predefined periods have been reached, the sum was stored and reset to zero so that it could continue for the next sample. The output was then fed to the FIFO memory where it was stored and later on saved on the SD card once the FIFO was half full.

B. PCB prototype

After following the guidelines and specifications outlined in section V, the obtained motherboard prototype is reported in figures 12, 13. Before proceeding with the manufacturing of the PCB a check was done using the 3D model to see if there was enough clearance between the Data Hub and the motherboard, as shown in figure 14, which resulted to be 0.55 mm. However, the actual distance was going to be more than that since the mating of the connectors was not taken into consideration in the 3D model. The motherboard was then manufactured with the available PCB mill at the student workshop mentioned in subsection II-B. The result without components soldered onto it is shown in figure 15.

Once the components arrived they were soldered onto the prepped board and the results are shown in figures 16, 17, and 18.

C. Testing

Once the board was fully assembled the testing phase started. The testing setup is shown in figure 19. The equipment used was a signal generator used to simulate the input into the ADC, an oscilloscope to double-check various voltages and signals, and lastly a power supply to power the board. A 2 MHz differential sine wave was generated with the signal generator and then fed into the ADC inputs. Different amplitudes were used to test the system, in particular 50 mV, 500 mV and 1 V peak to peak. The result of the ADC sampling of the aforementioned signal is shown in figure 20. In figure 21, a close up from figure 20 is shown. Due to the fast sampling rate of the ADC, it wouldn't have been possible to save all the samples at the same time. For this reason, it was necessary to save a series of samples every other second, to give enough time for the data to be saved successfully. These blocks of successive samples are visible in the enlargement in figure 21.

VII. DISCUSSION

A. ADC

The chosen ADC gives great flexibility in all aspects of the acquisition. These include the choice of voltage for input and output, the variable sampling frequency, and fully parallel output, giving the option to the developer if the need arises to use fewer bits for the acquired signal. The only downside of this choice is the footprint which is comparable to the one of the FPGA. The flexibility comes at the cost of the size

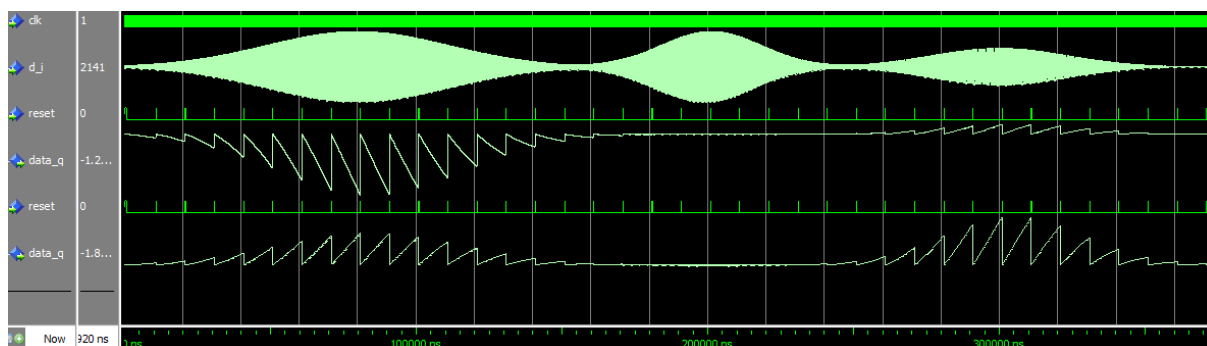


Fig. 11. Generated signals from testbench. The sampling clock is shown at the top, followed by the simulated input signal. Below are the I and Q results after each multiplication and accumulation. The jumps in these signals occur when the desired periods are reached, with the reset of the accumulator and data saving of the sample.

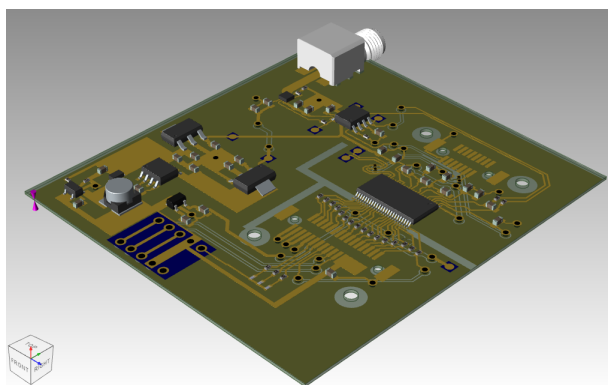


Fig. 12. Top view of motherboard prototype without Data Hub.

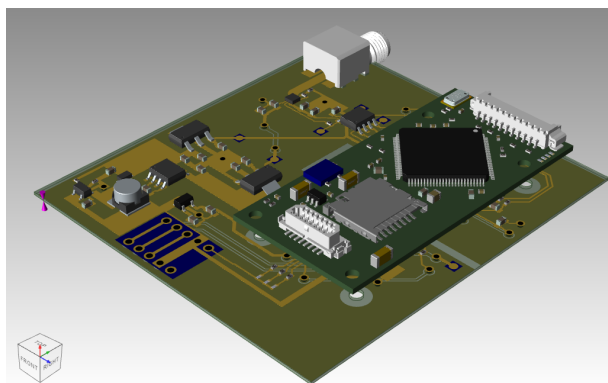


Fig. 13. Top view of motherboard prototype with Data Hub.

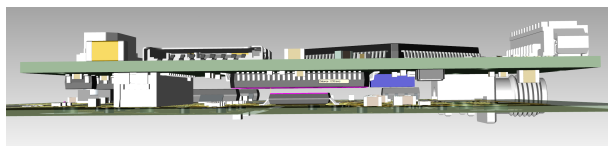


Fig. 14. Clearance between Data Hub and motherboard for stacked design.

and if it is needed due to not being able to fit the design in the final PCB, a new ADC might have to be investigated. Another good thing about this ADC is that it comes in different configurations, giving the possibility to increase the sampling

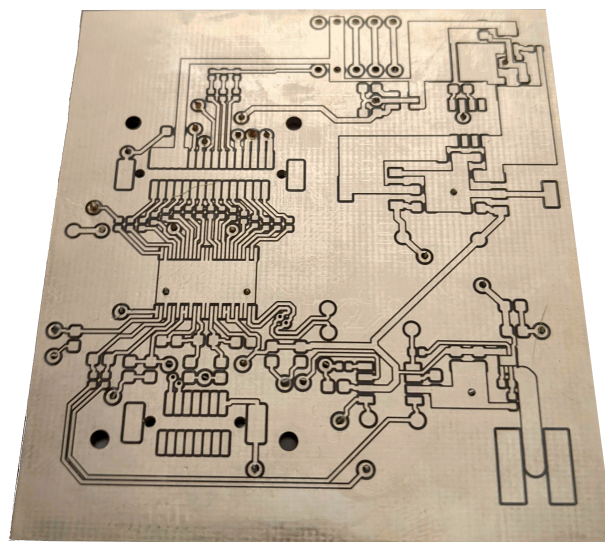


Fig. 15. Top view of manufactured PCB.

rate up to 80 MSPS just by changing the chip since the pin configuration is kept equal. From the tests performed so far the quality of the ADC is acceptable, except for some bits which report the wrong value. The causes of these errors are still unknown but could be caused by some settling time or rising time in the ADC, further testing is required.

B. PCB design

For this project, a 2 layers prototype was made, since it was possible to manufacture and test in-house. However, this design is not optimal and prone to noise. The reason for this is that some signal lines had to be made longer, to be able to go around other components. In a multi-layer design, this could be avoided by having planes for different voltages and signals but also ground planes resulting in better SNR and less space occupied.

After the components were soldered, an initial test of the board was performed. The voltages behaved as expected. Some problems were found with some components which needed to be changed and some sizes were not correct due to the availability of parts. The capacitors of the Switched-Capacitor

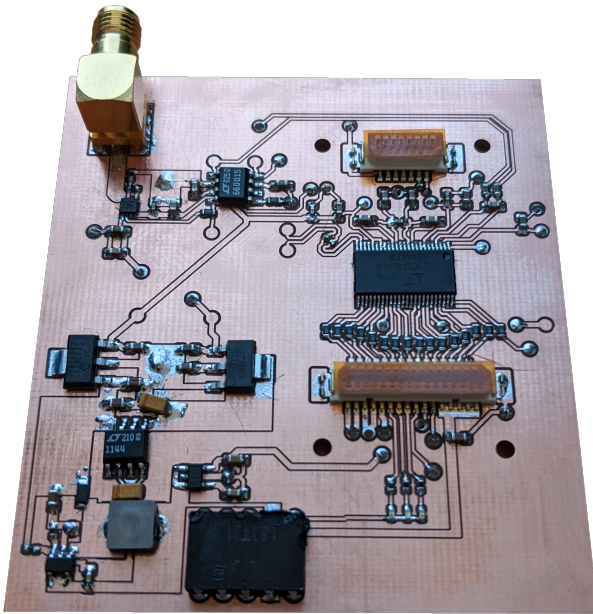


Fig. 16. Top view of motherboard prototype with soldered components without Data Hub.

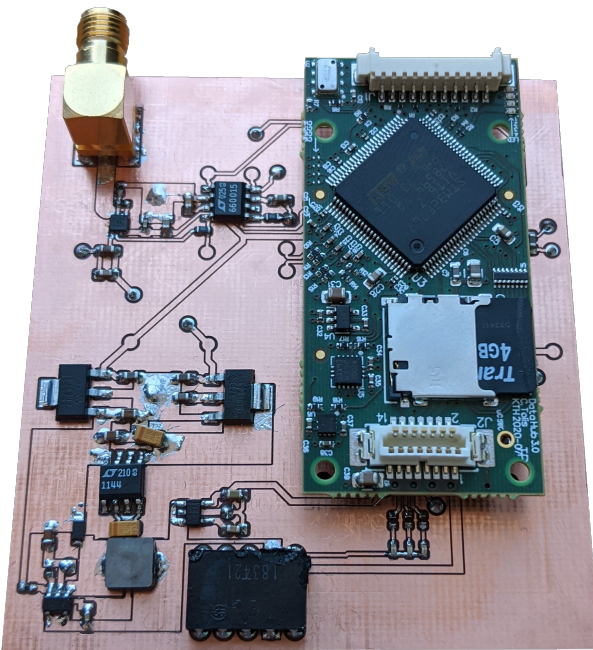


Fig. 17. Top view of motherboard prototype with Data Hub and soldered components.

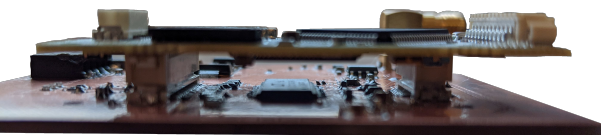


Fig. 18. Side view of prototype board with connected Data Hub.

Wide Input Range Voltage Converter LTC1144 had to be changed to a tantalum capacitor.

Another issue was found while testing. The problem is

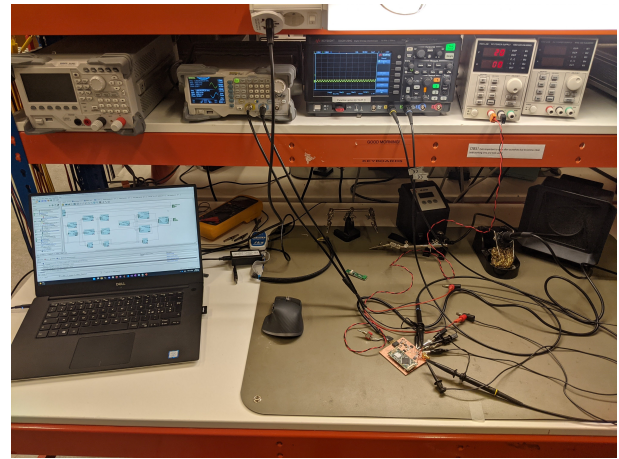


Fig. 19. Test setup used for evaluating the board. The equipment used are a signal generator, an oscilloscope and a power supply.

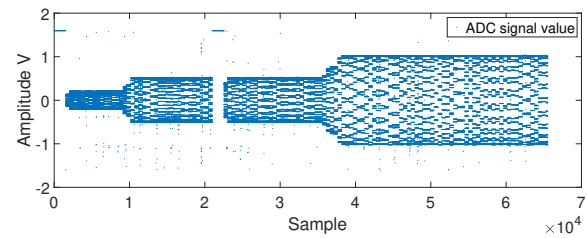


Fig. 20. Output of ADC recovered from SD card.

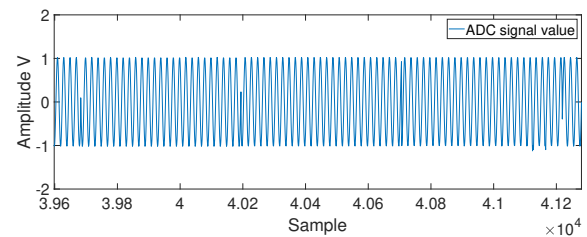


Fig. 21. Details of ADC output. Multiple blocks of successive saved samples are visible.

with the supply voltage to the first op-amp. The problem was quickly resolved but it needs some changes to the final design and it could have been easily avoided if another pair of eyes would have closely followed the design steps. For this reason, more testing and work are needed to be able to get real data.

C. Software

Comparing the results from the testbench and the MATLAB[®] illustration, the results behave as expected. However, to fully judge the quality of the solution, tests with the real hardware will have to be done. Furthermore, the saved data will have to be evaluated and compared with the simulations.

D. Future work

At the moment the sampling frequency and the predefined frequency are divisible, this makes it easy to perform the calculations and provides a base for the design. A complete version that handles non-divisible frequencies has to be made. Another aspect that hasn't been taken into consideration in this project is the design of the antenna, at the moment only the connector is provided and that could be investigated in a future project. After some initial testing, some issues were found with the PCB design which need to be addressed. Once these problems have been fixed, an analysis of the noise levels at the output and its quality shall be made. When this final step is complete a final version of the PCB in the form factor outlined in section III should be made.

VIII. CONCLUSION

After the research phase of the components, the ones that were deemed suitable were chosen. A prototype for the motherboard and connections was made, which once fully validated, could be reused for the final design. The implemented software gives the basis for other experiments to implement the same technique in other experiments. The chosen components give flexibility in multiple aspects, allowing the design to be implemented in a variety of configurations. In future work, all the remaining steps required to produce a wave propagation experiment on FPGA are outlined which would open the door to implementing this design for sounding rocket application for future REXUS experiments. In conclusion, there is still a lot of work to be done but this project provides a solid base for further development and testing before being able to qualify this system for a real-life application on sounding rockets.

APPENDIX A

MOTHERBOARD PROTOTYPE SCHEMATICS

APPENDIX B

MOTHERBOARD PCB DESIGN

APPENDIX C

SMARTDESIGN VIEW OF IQ CHAIN IMPLEMENTED IN THE LIBERO IDE.

ACKNOWLEDGMENT

The author would like to thank supervisor Nickolay Ivchenko for his outstanding help and expertise in multiple areas which showed me a broader look and inspired me greatly. This project wouldn't have been possible without the legacy hardware, software, and expertise from previous REXUS experiments who warmly welcomed me in my first year and taught me so many things throughout my education period. I would also like to thank my friends and classmates that were always there to support me, giving me new energies and drive. Most of all I would like to thank my family and girlfriend that were always there for me in the difficult moments.

REFERENCES

- [1] (2022, Apr.) REXUS | Rexus/Bexus. [Online]. Available: <https://rexusbexus.net/rexus/>
- [2] C. Lindstein. (2021, Sep.) Student rocket experiments | KTH. [Online]. Available: <https://intra.kth.se/2.844/centra/rymdcenter/studentrelaterat/stud>
- [3] (2022, May) T-minus dart. [Online]. Available: <https://www.t-minus.nl/space>
- [4] "IEEE Standard Definitions of Terms for Radio Wave Propagation," *IEEE Std 211-2018 (Revision of IEEE Std 211-1997)*, pp. 1–57, Feb. 2019, conference Name: IEEE Std 211-2018 (Revision of IEEE Std 211-1997).
- [5] B. Zolesi and L. R. Cander, *Ionospheric Prediction and Forecasting*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. [Online]. Available: <http://link.springer.com/10.1007/978-3-642-38430-1>
- [6] M. Friedrich, "Handbook of the lower ionosphere," Oct. 2016, publisher: Verlag der Technischen Universität Graz. [Online]. Available: <https://openlib.tugraz.at/handbook-of-the-lower-ionosphere-2016>
- [7] (2022, Apr.) Student Workshop. [Online]. Available: <https://www.kth.se/ee/spp/education/student-workshop/studentverkstan-1.683473>
- [8] (2022, Apr.) PCB Mill. [Online]. Available: <https://www.kth.se/ee/spp/education/student-workshop/verkstan/pcb-fras-1.964779>
- [9] C. Tolis, "The data hub pcb," KTH Royal Institute of Technology, Stockholm, Tech. Rep., Feb. 2020.
- [10] (2022, Apr.) ProASIC3 | Microsemi. [Online]. Available: <https://www.microsemi.com/product-directory/fpgas/1690-proasic3#proasic3-e>
- [11] R. Oshana, "4 - Overview of Digital Signal Processing Algorithms," in *DSP Software Development Techniques for Embedded and Real-Time Systems*, ser. Embedded Technology, R. Oshana, Ed. Burlington: Newnes, Jan. 2006, pp. 59–121. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780750677592500065>
- [12] Team PRIME, "PRIME student experiment description," KTH Royal Institute of Technology, Stockholm, Internal KTH REXUS document, Nov. 2019.
- [13] "Bus Speed (Default Speed/High Speed/UHS/SD Express) | SD Association," Dec. 2020. [Online]. Available: <https://www.sdcard.org/developers/sd-standard-overview/bus-speed-default-speed-high-speed-uhs-sd-express/>
- [14] N. Blaunstein and C. G. Christodoulou, *Ionospheric Radio Propagation*. New Jersey: John Wiley Sons, Ltd, 2014, pp. 591–638.
- [15] (2022, Apr.) LTC1744 Datasheet and Product Info | Analog Devices. [Online]. Available: <https://www.analog.com/en/products/ltc1744.html#product-overview>
- [16] "LT6600-15 - Very Low Noise, Differential Amplifier and 15MHz Lowpass Filter," p. 12, May 2022. [Online]. Available: <https://www.analog.com/media/en/technical-documentation/data-sheets/660015fb.pdf>
- [17] (2022, Apr.) OPA858-Q1 data sheet, product information and support | TI.com. [Online]. Available: <https://www.ti.com/product/OPA858-Q1#tech-docs>
- [18] (2020, Sep.) 11 Best High-Speed PCB Routing Practices. [Online]. Available: <https://www.protoexpress.com/blog/best-high-speed-pcb-routing-practices/>
- [19] T. Hubing, "PCB EMC design guidelines: a brief annotated list," in *2003 IEEE Symposium on Electromagnetic Compatibility. Symposium Record (Cat. No.03CH37446)*, vol. 1, Aug. 2003, pp. 34–36 vol.1.
- [20] B. C. Wadell, *Transmission Line Design Handbook*. Artech House, 1991, google-Books-ID: MyxTAAAMA AJ.
- [21] "Libero® SoC Design Suite Versions 2022.1 to 12.0 | Microchip Technology," Apr. 2022. [Online]. Available: <https://www.microchip.com/en-us/products/fpgas-and-plds/fpga-and-soc-design-tools/fpga/libero-software-later-versions>
- [22] M. Tsamsakizoglou, "Radiation tolerant satellite communication modem," Master's thesis, KTH, Space and Plasma Physics, 2012.

Building A Fixed Wing Autonomous UAV

Erik Barsby and Casper Augustsson

Abstract—The goal of this bachelor thesis has been to evaluate and test the available open source software and commercial hardware for potential later use as the electrical system in the ALPHA UAV. ALPHA is a student project, with the goal of building an autonomous drone capable of high altitude, long-endurance missions to gather data from electromagnetic phenomena in the atmosphere. Data later to be used in research at the facility of Space and Plasma physics at KTH. The evaluation has been done by constructing of an MVP, to prove that the open source software and commercial hardware can be used to build an autonomous UAV.

Sammanfattning—Målet med denna kandidatuppsats har varit att evaluera och testa öppen källkod tillsammans med kommersiell hårdvara för att potentiellt kunna nyttjas som elektriskt system i ALPHA UAV. ALPHA UAV är ett studentprojekt, med målet att bygga en autonom drönare kapabel att genomföra höghöjdsflygningar med lång uthållighet för att kunna samla in data från elektromagnetiska fenomen i atmosfären. Data som senare kan nyttjas i forskningssyfte på institutionen för rymd-och plasmafysik på KTH. Evalueringen har gjorts genom att konstruera en MVP, för att bevisa att öppen källkod och kommersiell hårdvara kan nyttjas för att bygga en autonom UAV.

Index Terms—fixed wing, bachelor thesis, KTH, Ardupilot, autopilot, electric propulsion, UAV, autonomous.

Supervisor: Mykola Ivchenko

TRITA number: TRITA-EECS-EX-2022:159

I. INTRODUCTION

Building an unmanned aerial vehicle (UAV) from scratch is not an easy task, the margin of error is extremely small and therefore requires not only accuracy but also the courage to make the UAV airworthy. Today, there are plenty of options to build a sophisticated UAV with the help of open-source software and hardware. In 2019, the student project ALPHA started at KTH with the purpose of building an autonomous aircraft that will have the ability to measure light phenomena in the upper atmosphere. To obtain sufficiently good sensor data, the UAV must fly at an altitude of a maximum of 15 km for long periods, which poses high demands on both software and hardware that must enable safe flight in all circumstances. The aim of this bachelor thesis has been to evaluate and test the available open source software and hardware for potential later use as the electrical system in the ALPHA UAV. To test the potential hardware and software a fixed wing UAV has been built, where the UAV has undergone several flight tests to prove that the hardware and software are capable of safe flight.

A. Abbreviations

BEC	Battery Eliminator Circuit
BLDC	Brushless Direct Current
CCW	Counter Clockwise
CG	Center of Gravity
CW	Clockwise
DC	Direct Current
EPO	Expanded Polyolefin
ESC	Electronic Speed Controller
FBWA	Fly By Wire A
FC	Flight Controller
GCS	Ground Control Station
GPS	Global Positioning System
I2C	Inter Integrated Circuit
LiPo	Lithium Polymer Battery
MVP	Minimum Viable Product
PWM	Pulse Width Modulation
RC	Radio Controlled
RPM	Revolutions Per Minute
UART	Universal Asynchronous Receiver Transmitter
UAV	Unmanned Aerial Vehicle

B. Background

The goal of the ALPHA project is to build a UAV with the purpose of data collection at high altitudes. This data will later be used in research at the Division of Space and Plasma Physics at KTH [1]. To achieve this, the following core requirements need to be met:

- The UAV shall be suitable for cruise at altitudes of between 5 km and 15 km, at speeds of 50 km/h to 200 km/h
- The endurance shall be of at least 6 hours.
- The payload/control compartment shall provide a volume for easy mounting of the payload, power and control systems, and easy connection of the propulsion motor and servomotors.
- The UAV should fly autonomously.
- The payload and control compartment shall be open upward, for easy implementation of the zenith-looking optics.

The ALPHA project and this bachelor thesis do not have the same requirements, where above requirements are for the ALPHA project.

C. Aim

The main scope of this bachelor thesis is to implement an open source hardware/software platform, on an arbitrary fixed wing fuselage and make it fly autonomously. To create an electronic and software setup that that will lay the backbone of the electronics and software system in ALPHA UAV. Much

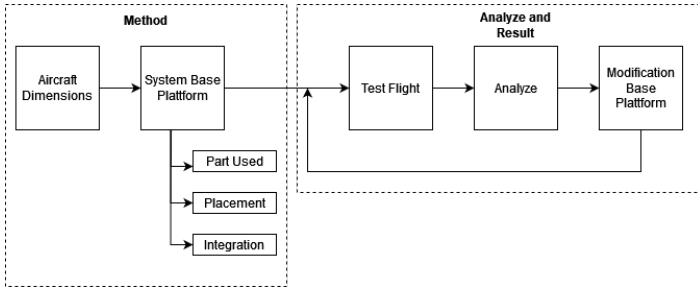


Fig. 1. Workflow of the Method, Analyze and Result

effort has been put into making the setup independent of the fuselage due to the design of the ALPHA UAV fuselage not being completely determined. All the knowledge gained during the project also serves as feedback for the ongoing construction of the ALPHA UAV. The goals of this thesis are:

- Choose what electronics and software are needed to build an autonomous UAV
- Build the UAV by integrating the chosen electronics and software systems into a fuselage
- Learn how to operate the UAV and make it fly autonomously

D. Previous studies

Many UAVs have been built by the community running Ardupilot. And several scientific articles use Ardupilot as a base for research and development, dependent on having a working UAV platform [2] or using it as a codebase for flight control development [3]. But few articles describe, from a system perspective, the implementation of a fixed wing UAV using Ardupilot and available hardware. Benoît Henrivaux did a similar thesis in subject and content as this bachelor thesis 2017. In [4], Henrivaux goes through how to integrate Ardupilot into a fossil fuel RC-UAV. The main difference between Henrivaux thesis and this article is that this article describes electric propulsion together with an upgraded version of the hardware and software.

E. Disposition

In this section the disposition of this report is explained, containing theory, method, result and analysis, discussion and conclusion. In theory, all the necessary background information needed during the project is presented, including the function of all the subsystems used in the UAV. In method the specific components used in each subsystem are presented, to be followed by how all the subsystems were integrated. In the result and analysis, the performed flight tests are gone through and analyzed. In discussion, the major learning outcomes from the flight tests are summarised and described. Figure 1 shows the workflow used during the project. What is important to point out is that the workflow throughout the project has been agile, denoting that the project did not have a strong distinction between the classical method, analysis and result. Instead, they were merged and used together to create a workflow based on iteration to produce a functional UAV.

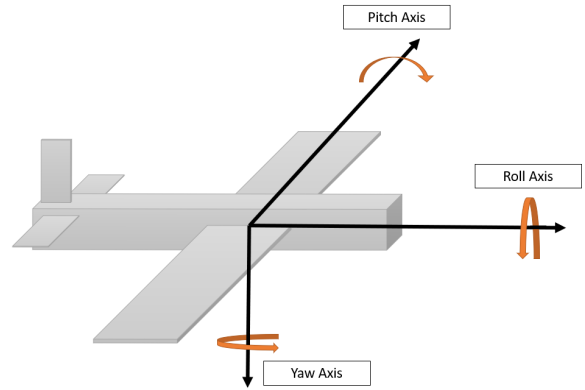


Fig. 2. Roll, pitch and yaw explained in picture

II. THEORY

This section go through the theory used in this project.

A. Mechanics

1) *Center of lift*: The aircraft is subject to a force of lift. All the lifting force from the entire wing can be added up to one force vector. The position on the wing where this force vector is centered is called center of lift [5].

2) *Center of gravity*: The force of gravity is acting on every part of the entire plane. This fact tells us that there is a point on the plane that if balanced at, all the torques on that point from the plane cancel out. This point on the plane is called center of gravity [5].

3) *Landing gear*: The purpose of the landing gear is to make a safe and stable takeoff and landing easier, but also to give the plane the ability to start and take off on the ground without damaging the fuselage. The landing gear also serves to give the aircraft more propeller clearance. The landing gear must ensure the stability of the aircraft and is done by having the center of gravity well within the area between the upholding points. The upholding points are often wheels and should for stability be as big and wide as possible.

4) *Roll, Pitch and Yaw*: Roll, pitch and yaw are the directions in which an aircraft is commonly controlled [5]. In Figure 2 roll, pitch and yaw are illustrated.

5) *Control surfaces*: Rotation in roll, pitch and yaw is common, on fixed wing planes, achieved by control surfaces on the wings. The air passing over and under the wings creates a force on the control surfaces, which creates torque on the plane. Depending on the direction of the control surfaces and which control surfaces which are used the pitch, roll and yaw can be controlled [6]. The placement of the control surfaces is shown in Figure 3. Looking at the plane from behind, roll, pitch and yaw can be controlled by using the control surfaces according to Table I.

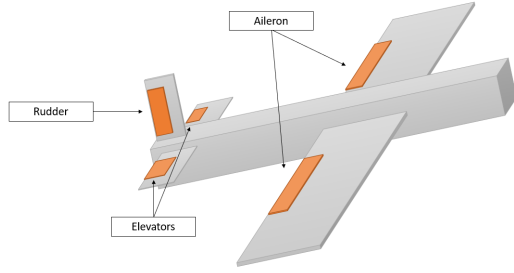


Fig. 3. Common control surfaces of fixed wing plane, source [7]

TABLE I
CONTROL SURFACE POSITION TO ROLL, PITCH AND YAW TABLE
N: NEUTRAL, R: RIGHT, L: LEFT, CW: CLOCKWISE, CCW: COUNTER
CLOCKWISE

	Aileron R	Aileron L	Elevator R	Elevator L	Rudder
Roll CW	Up	Down	N	N	N
Roll CCW	Down	Up	N	N	N
Yaw R	N	N	N	N	R
Yaw L	N	N	N	N	L
Pitch Up	N	N	Up	Up	N
Pitch Down	N	N	Down	Down	N

6) *Propeller*: Propellers for RC aircraft come with two dimensions given in inches. The first one is called arc diameter. This number gives the diameter of the circle that the propeller turns in when spinning. The second number is called pitch. Pitch gives a theoretical number of how far the propeller will travel forward in the air if it spins one revolution. The pitch determines the inch per revolution the propeller theoretically will travel. Both of these are important and need to be matched with the motors and fuselage [8]. Big propellers with too much traction is heavier to accelerate for the motors and give rise to more unwanted wind effects on the fuselage, as the slipstream effect seen in Figure 4.

7) *Motor placement*: The motors should be placed to create thrust in the desired direction of flight without creating torque on the fuselage. Torque from the motor on the fuselage can be due to other reasons than the motor not acting in desired flight direction. It can also be due to the torque applied to accelerate the propeller. This torque also acts upon the fuselage, giving rise to roll. If the plane is propelled by one motor, because the propeller is turning in only one direction, both the air coming into and out from the propeller will also spin in one direction. This air will vortex around the fuselage onto the wings and give rise to a torque on the flight body giving rise to roll and yaw. This effect is called the slipstream effect [9], the effect is illustrated in Figure 4. To counter these two sources of undesired roll and yaw, due to the asymmetries from having one propeller spinning in one direction, the propeller can be offset by a design dependent amount of degrees to create a throttle dependent counteracting roll and yaw torque.

B. Electronics

1) *Flight controller*: A flight controller (FC) is a circuit board with a variation of complexity depending on the application of use. Because there are several different kinds of

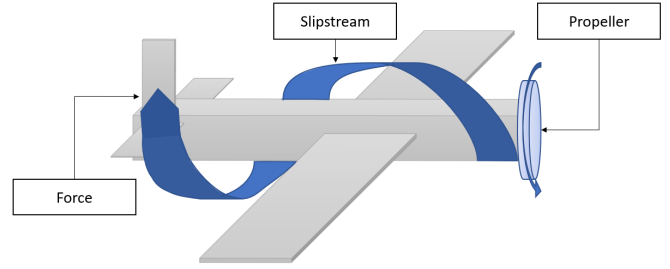


Fig. 4. Slipstream effect

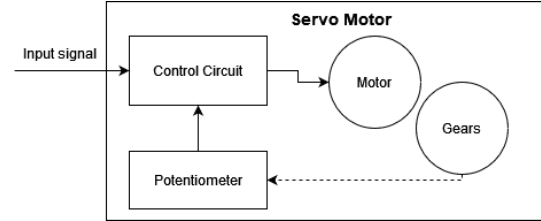


Fig. 5. Generic servo motor

controllers, a set of categories explained in [10] has been established to describe the different features of flight controllers:

- **Sensing**: The flight controller consists not only of computational electronics but also contains sensors. The core ones are accelerometer, gyroscope, barometer, and compass. The accelerometer and the gyroscope measure the linear and rotational acceleration of the aircraft. The compass measures the direction and the barometer estimates the altitude from the air pressure.
- **Controlling**: One of the main features of the flight controller is to have the ability to control the UAV with the help of the sensor onboard or by direct control from a pilot.
- **Communicating**: Communicating is the ability to send and receive information with systems onboard the UAV and remotely to systems far away from the UAV.

2) *Servo motor*: A servo is an electrical motor that can be moved to a specific position with the help of internal feedback [11]. A generic design of a servo motor is shown in Figure 5. The input signal to the servo gets decoded by the control circuit and the decoding results in a signal to the DC-motor that moves the servo gears to a position. The potentiometer works as feedback to the control circuit to indicate the position of the gear.

3) *Telemetry*: Telemetry is the ability to measure and collect data at a distant point and communicate it back to a receiving unit for observation [12]. In a UAV application, telemetry devices are usually connected to the flight controller, and communication is carried out with the use of radio to the receiving unit. As shown in Figure 6 the receiving unit is connected to a ground control station (GCS) that provides the interface to human control.

4) *Radio, receiver and transmitter*: In addition to the UAV flying completely autonomously, it is also usually possible to

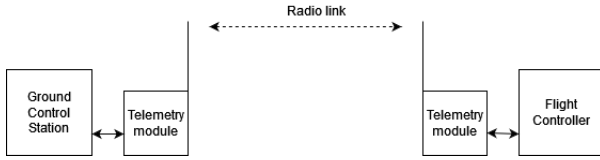


Fig. 6. Generic telemetry module setup

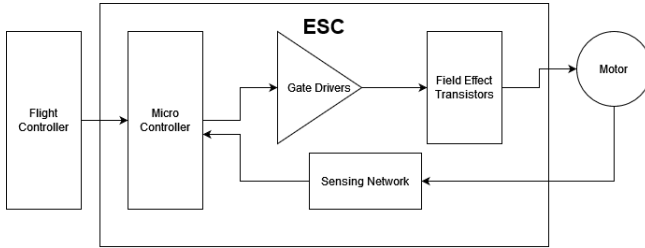


Fig. 7. Generic ESC

fly an UAV using a radio transmitter. The radio transmitter works at a fixed frequency band, commonly 2.4 GHz [13] and can send commands to the receiver onboard the UAV. The receiver decodes the signal and passes it to the flight controller for further processing and control.

5) *GPS*: Global Positioning System (GPS) is a radio navigation system that can determine the user's position, latitude, longitude, and altitude on earth [14]. This is done with the help of several orbiting satellites that transmit a modulated carrier wave with information to an arbitrary receiver on earth. By using four satellites, a receiver has the ability to calculate its position, ground speed, and direction.

6) *ESC and BLDC Motor*: An ESC is an electronic device that has the ability to control and adjust an electrical motor [15]. The main parts of a generic three phase ESC are described in Figure 7:

- **Micro controller**: The microcontroller works as the interface between the flight controller, gate drivers, and sensing network. At a given input signal, the microcontroller reads the sensor data and activates the gate drivers. More sophisticated ESC allows configuration of the microcontroller, hence enabling deeper customization of the system.
- **Gate Drivers**: To be able to drive the field effect transistor gates, the gate drivers receive the signal from the microcontroller and amplify it.
- **Field Effect Transistors**: Field effect transistors are used to drive the right amount of current through the coils in the BLDC motor and are toggled by the gate drivers.
- **Sensing network**: The sensing network is either made up of using the counter electromotive force from the motor or using a Hall-effect sensor in the stator. Both are used to sense the position of the rotor and send information back to the microcontroller.

A Brushless Direct Current (BLDC) Motor is an electrical synchronous motor that is made up of a stator with windings and a rotor with permanent magnets. During operation, the

windings get energized when the field effect transistors toggle. By energizing the windings at the right time, a magnetic field will be created, attracting the permanent magnets in the rotor. Changing the frequency of which field effect transistor toggles will lead to a changed speed of the rotor. The most common BLDC motor parameters in commercial use can be categorized as:

- **Length and Height**: Different applications require different kinds of motors, where the length and height will affect the torque it produces [16].
- **KV**: KV-rating refers to the number of revolutions per minute (RPM) a motor does per applied volt [16].
- **Max Power**: Defines the max amount of power the motor can withstand during operation.

7) *LiPo-Batteries*: Lithium-ion polymer battery (LiPo) is a technology using polymer electrolyte and lithium metal. The cells in the batteries have a high energy density and discharge rate which makes them ideal for use in low weight high power applications [17]. The cell voltage is dependent on design and charge, but a general guideline of nominal voltage is approximately 3.6 V per cell.

8) *Power Module and Battery Elimination Circuit*: A Power Module is a system that powers the flight controller and monitors the voltage and current from the battery [18]. Battery Elimination Circuit (BEC) is a DC/DC voltage regulator used to provide subsystems (servos, sensor, LED, etc.) with the right amount of power. This is done by using a buck-converter configuration that can step down a higher voltage to a lower voltage with high efficiency [19].

C. Protocols

1) *PWM*: PWM or Pulse Width Modulation is a widely used modulation method to create an average analog voltage, current or power delivery with digital electronics [20]. One feature of digital electronics is that the voltage is either on or off. This is limiting when driving analog electronics, for example a DC motor. By varying the voltage applied to the DC motor, the speed can be varied. This can be achieved with digital electronics by switching the power on and off many times faster than the magnetic rotation cycle of the DC-motor. This way, the electric inertia of the motor will make the motor experience a voltage equal to the time mean of the switching digital signal. By varying the time the digital signal is switched on and off, the time mean can be varied, and this way an artificial analog signal can be created with digital electronics. This is shown in Figure 8.

2) *UART Serial*: UART Serial is a full duplex communication protocol where two devices send messages to each other at the same time [21], often used between two devices that need to share information. UART needs 3 cables and only allows communication between two devices in one line.

3) *I²C*: I²C is a protocol often used to connect multiple sensors to a micro-controller. I²C is half duplex, which means sensors can both send and receive packages, but not at the same time. The advantage of I²C is that one micro-controller

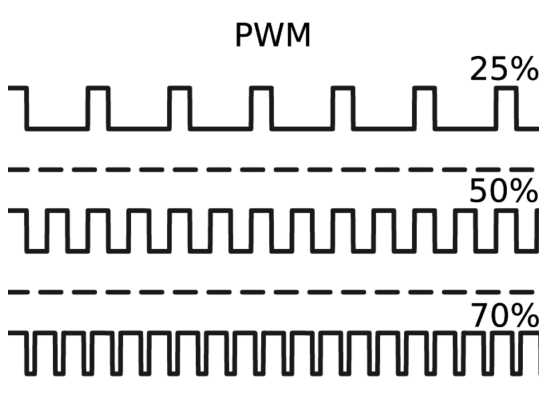


Fig. 8. Pulse width modulation labeled with corresponding percent of supplied max voltage

can connect up to 127 devices with only 3 cables, 2 signal cables, and ground. The speed of the standard implementation is 100 kbit/s [22].

D. Software

1) *Ardupilot*: Ardupilot is an open source flight computer software that can be run on multiple hardware platforms. It is intended to be used for radio controlled (RC) UAVs, copters, submarines, and rovers. Ardupilot is optimized to be extremely versatile and work with almost any vehicle construction imaginable out of the box and has the following features:

- **Stabilization**: Enables the operator to give high level instructions to vehicles on how they should move, in contrast to the traditional way of controlling the motors and control surfaces.
- **Autopilot**: Enables the operator to program the vehicle where and how it should move and then the vehicle can do it autonomously without the control of an operator.
- **Telemetry**: Ardupilot can exchange data with the GCS during flights. This enables the operator to change autonomous flight path during missions and give continuous updates on the state of the aircraft.
- **Datalogging**: Ardupilot logs sensor and state data during operation. This feature enables the operator to analyze the flight afterward and helps to conclude the event of errors.
- **Sensor and hardware library**: Ardupilot has a builtin support for a wide range of sensors and hardware that is connected by I^2C , $UART$ —*serial*, and many other hardware interfaces. This makes it easy to connect commercial and easy-to-use products for a variety of applications.
- **Failsafe options**: Ardupilot has support for various failsafe options, the core one being: return to the launch site if the connection to the operator is lost.

Ardupilot has a wide list of flight modes available for fixed wing aircraft [23]. The core ones are the following:

- **FBWA**: In this mode the operator gives high-level commands to the aircraft with the RC- transmitter II-B4, commands such as alternating the roll, pitch, and yaw angles. Ardupilot, using control theory, continuously change

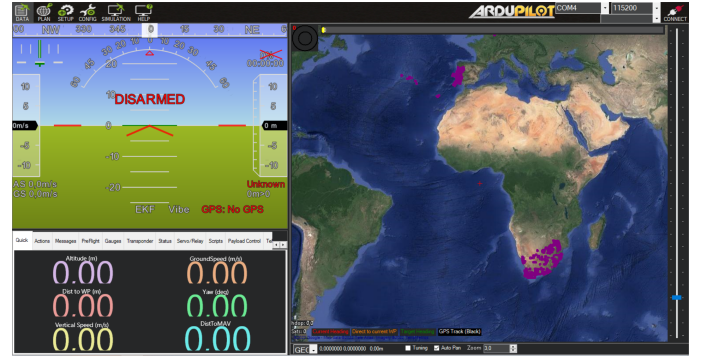


Fig. 9. Mission planner interface

the control surface positions to make sure that these angles are sustained. The FBWA modes do not control the motors of the aircraft. Instead, the operator has full control over the motors [23].

- **Manual**: In this mode the operator has direct control of the motors and control surfaces.
- **Loiter**: Ardupilot will autonomously fly at a determined altitude, speed, and radius around the point in which the UAV was at when entered loiter mode.
- **Auto**: Ardupilot will autonomously fly according to a planned path. A path is made up of several waypoints, consisting of latitude, longitude, and altitude.

2) *Mission Planner*: Mission planner is an open source software and can be used for various types of aerial, ground, and underwater vehicles. The main features, as explained in [24], are:

- **Ground control station**: Mission planner can act as the interface to a telemetry module and as a dynamic control supplement for monitoring the UAV. As seen in Figure 9, the mission planner interface consists of a map showing the GPS status of the vehicle and gauges displaying vital information.
- **Configure autopilot**: Mission planner can be used to load firmware into the autopilot system and be used to setup, configure and tune the vehicle.
- **Analyze log files**: Apart from monitoring the data sent by the telemetry module, mission planner can be used to collect and save data during flights [25]. After flight, these log files can be used to analyze the behavior of the autopilot and vehicle dynamics.
- **Mission commands**: To control and change vehicle behavior during flight, mission planner and a telemetry module can be used to send commands to the vehicle.

III. METHOD

This section explains what specific parts were used during the project and how they were integrated.

A. Aircraft Dimensions

As explained in the I-C the focus of this project has been on implementing a hardware/software solution for a UAV. The entire fuselage is an off-the-shelf solution, which means that

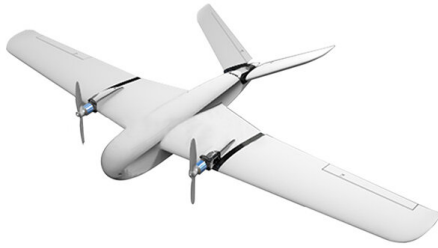


Fig. 10. X-UAV Clouds

the project has not been focused on studying or building the fuselage from scratch. The fuselage used in the project is called X-UAV Clouds and is shown in Figure 10. The UAV is an EPO fuselage with a twin motor configuration and a wingspan of 1880 mm. Were the specifications are:

- General specification:
 - Material: EPO
 - Overall Length: 960mm
 - Height: 260mm
 - Tail configuration: V-Tail
 - Mission Payloads: 60-1200g
 - Cruising Speed: 50-80 km/h
 - Cruise Time: 1.5-3h
- Recommended Parts:
 - Motor: 2814 KV840 x 2
 - ESC: 40A x 2
 - Blade: 11 x 7

B. System Base Configuration

1) *Parts Used:* Many parts were available at KTH. Some parts did come from previous projects, others were bought from the maker spaces available at KTH. This made the process of getting components much easier. The goal of this thesis was not to make an efficient long endurance UAV, many parts were therefore chosen because they were good enough and available, not because they were optimal.

a) *Mechanics:*

- **Motor:** The system was fitted with 2 BLDC motors, one on each wing. They were configured in such a way that they rotate in the opposite direction. This is to ensure that their torque on the fuselage cancels each other out. The motor specification was decided to be as similar as possible to the ones recommended by the manufacturers of the fuselage. Turnigy aerodrive 1050 KV was chosen.
- **Propeller:** The propellers were chosen to be 11x7, this was recommended by the manufacturers of the fuselage III-A.

- **Control Surfaces:** The control surfaces of the Clouds UAV are the following:

- Aileron
- Elevator

As seen in Figure 10, the fuselage doesn't have any rudder as shown in 3. Instead, it has its Elevators configured in a V-tail configuration. This way, one less control surface is needed. If the elevators turn in the opposite direction to each other they will act as a rudder [26].

- **Landing gear:** The aircraft was configured with one pair of wheels in front of the center of gravity and one sled at the tail of the aircraft, to ensure a safe start and landing as explained in II-A3.

b) *Hardware:*

- **Power Module:** The low voltage electronics of the aircraft are powered by the main battery. But to lower the voltage and ensure stable operation, a no-brand power module was chosen. This module also contains sensors that measure the current drawn from the battery and the voltage across the battery. This data is sent to the GCS and used to inform the operator of the remaining endurance.
- **BEC:** Many ESCs on the commercial market have an integrated BEC to enable power to the UAVs servos. The ESC chosen for this project didn't have an integrated BEC and therefore a standalone solution was chosen. The BEC is made of a DC/DC buck converter of the model LM2595 [27], with the output voltage of 5.7 V to supply the servos. As explained in section II-B8, the buck converter was chosen because of its high efficiency.
- **Battery:** Two identical LiPo batteries were chosen as the main power supply onboard the UAV with 6000 mAh of capacity and 5-cell technology. Beyond the voltage and capacity specification, the LiPo was also chosen because of its low weight to capacity ratio and high discharge rate as explained in II-B7.
- **Flight Controller:** Pixhawk 2.4.6 [28] was chosen because it had many connections for sensors and was fully Ardupilot compatible. The flight controller is shown in Figure 11 and 12.

The Pixhawk 2.4.6 has the following specifications:

- Processor:
 - 32-bit ARM Cortex M4 core with FPU
 - 168 MHz/256 KB RAM/2 MB Flash
 - 32-bit failsafe CO-processor
- Sensors:
 - MPU6000 as main accelerometer and gyroscope
 - ST Micro 16-bit gyroscope
 - ST Micro 14-bit accelerometer/compass (magnetometer)
 - MEAS barometer
- Power:
 - Ideal diode controller with automatic failover
 - Servo rail high-power (7 V) and high-current ready
 - All peripheral outputs over-current protected
 - All inputs ESD protected

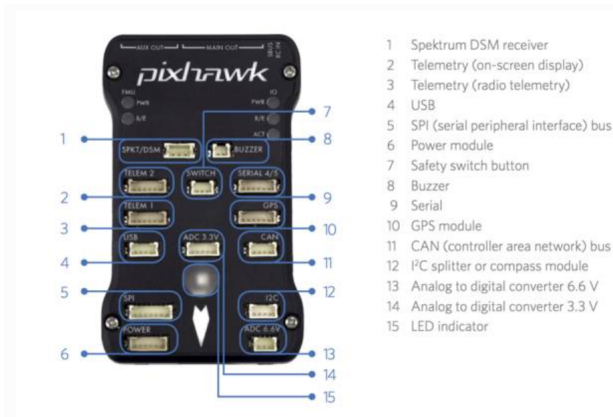


Fig. 11. Picture of Pixhawk and main outputs

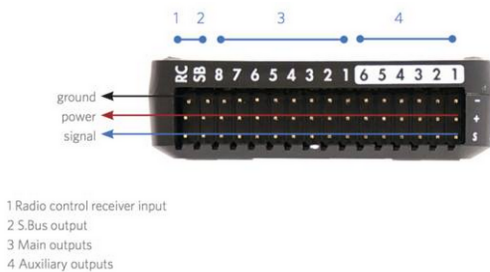


Fig. 12. Picture of Pixhawk front panel

– Interfaces:

5x UART serial ports, 1 high-power capable Spektrum DSM/DSM2/DSM-X Satellite input
Futaba S.BUS input
PPM sum signal
RSSI (PWM or voltage) input
I2C, SPI, 2x CAN, USB
3.3V and 6.6V ADC inputs

– Dimensions:

Weight 38 g
Width 50 mm
Height 15.5 mm
Length 81.5 mm

- **ESC:** The ESC that were chosen for this project was the 45A BLHELI-32. These ESC were chosen primarily because of the ability of deeper configuration of the microcontroller as mentioned in II-B6. Because two BLDC motors were used, two identical ESC had to be used with the same configuration. It is essential to have the same configuration of the ESC. If there is the slightest difference between the two, the motors will pull unevenly.
- **GPS:** GPS module Ublox M8N was used. UBlox M8N was chosen because it was available in a weather proof casing and is compatible with Ardupilot. It is a decent GPS module working according to II-B5. Full specifications can be found in [29].
- **Telemetry module:** Two SIK radio modules were used for telemetry. One is connected to the flight controller

and one connected to the GCS. SIK radio is an open source software platform that is intended to be run on telemetry modules used in UAV applications. They were used because they are easy to use and have a range of a couple of kilometers [30].

- **Radio transmitter and receiver:** FrSky Taranis transmitter was used together with a FrSky X8R receiver to control the aircraft. As mentioned in II-B4 it operates at 2.4 GHz and sends commands directly to the FC. It is a high-end, handheld RC-transmitter with many control options. FrSky Taranis was used because of its wide support of external transmitter modules and therefore flexibility to change to different communication protocols at different frequencies [31].
- **Servo:** To control the control surfaces, as explained in II-A5, 4 servo motors were used. These servos use metallic gears with a torque of 0.17 kgm. The servos need to have metal gears and high torque since the forces on the control surfaces can be high during heavy loads.

c) Software:

- **Windows:** To install and configure Ardupilot on the flight controller, the GCS Mission Planner is needed. Mission Planner is designed to run only on Windows 7 and later. Therefore, a computer running Windows 7 or later is needed.
- **CH340 driver:** Many of the electronics which were used contain the CH340 chip for USB connectivity. To make a computer able to recognize these devices, the CH340 driver is needed. The drivers can be found on various websites.
- **Mission Planner:** For GCS Mission Planner is used. There is also other ground control station software compatible with Ardupilot, but Mission Planner was used because it is easier to configure and tune Ardupilot with compared to the alternatives.

2) **Placement:** All the physical equipment explained in section III-B1 were position in the UAV and is shown in Figure 13 and 14. Motors, servos, and GPS had a predefined slot in the aircraft, all other equipment were placed according to a specific function. The telemetry module (VII in 13) was placed as far away as possible from the GPS (II in 13) to avoid interference. The battery was placed in the nose to get the CG align with the center of lift, explained in II-A2 and II-A1. The ESC was placed closed to the motors to avoid the unnecessary length of power cables and the FC in the center to act as the core of the electronics.

3) **Integration:** Integration shows how all the systems were connected and how the hardware/software were configured.

a) Connection:

Figure 15 shows how all the hardware of the UAV is connected electrically. The main part of the system is the flight computer, Pixhawk, on the front of the Pixhawk all the sensors are connected 11. The connection between Pixhawk, servo motors and ESC is done through the connections at the upper side 12. Each column of pins in 12 corresponds to one output connection including power delivery.

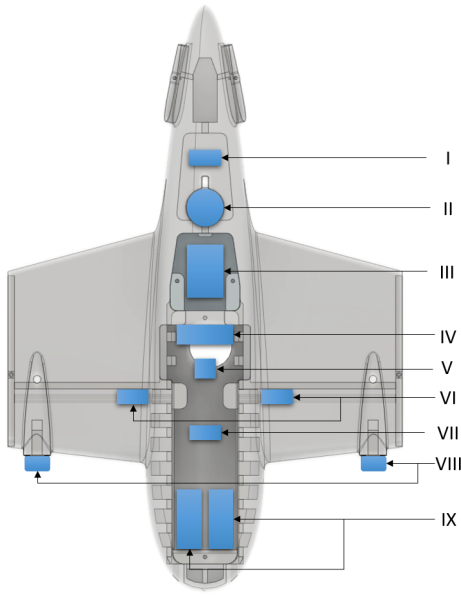


Fig. 13. I:Radio Receiver, II:GPS, III:Flight Controller, IV:BEC, V:Power Module, VI:ESC, VII:Telemetry, VIII:motors and IX: Batteries

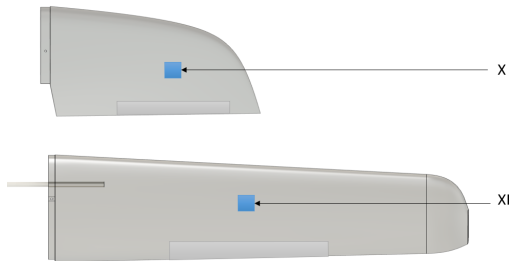


Fig. 14. X:Servo V-tail and XI: Servo Aileron

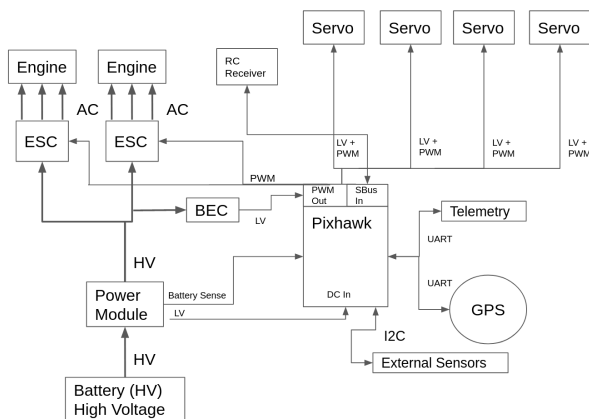


Fig. 15. Picture of electrical system. HV:High Voltage, LV:Low Voltage

b) Configuration:

The configuration of the hardware and software mainly describes how the Ardupilot software is tuned and customized to be able to function together with the hardware. These setups and calibrations are the baseline of the UAV configuration:

- **Flashing FC:** To be able to flash the FC, Mission Planner had to be downloaded. The next step was to flash the FC with the appropriate firmware, in this case UAV V4.1.7 was used. This was done using the online guide [32].
- **Calibration and Radio Setup:** The first step was to bind the radio receiver with the transmitter. To bind the transmitter and receiver, Taranis manual was used [33]. The radio transmitter was set to be able to output 6 different RC channels to the FC, these were:
 - Channel 1: Aileron
 - Channel 2: Elevator
 - Channel 3: Thrust
 - Channel 4: Rudder
 - Channel 5: Arming
 - Channel 6: Flight Mode

To check if the channels from the transmitter matched the channels of the FC the Radio-Calibration tab in Mission Planner was used. To calibrate the radio the same Radio-Calibration tab was used in Mission Planner together with the online guide [34].

- **Setup Telemetry:** Ardupilot use telemetry by default if a telemetry module is connected to the FC. Before the telemetry modules can be used, they need to be set to communicate with each other. This is easiest done in Mission Planner. Before starting make sure you have the CH340 driver installed III-B1c. How to configure them in detail is described in [30].
- **Setup Servo Channels:** To be able to command the control surfaces of the UAV, the FC needs to map the radio channels to the correct servo connected to the corresponding control surface. This was done using the online guide [35] and by knowing the connections of the servos to the FC. Important to note, that the specific FC determines how the servos should be connected and configured to work properly.
- **Setup Failsafe** Failsafe is initialized when Ardupilot realizes it lost contact with the operator. Because many ground tests were performed, with the propellers mounted, it was decided to set the failsafe to FBWA with neutral pitch, roll, and yaw for safety reasons. This means that Ardupilot will not give any throttle when contact is lost with GCS or the RC transmitter and only try to keep the aircraft stable while slowly descending to the ground. How to configure failsafe is described in [36]. It is crucial to configure failsafe and to know how it works, otherwise, there is a great risk that people get injured during the operation of the UAV.
- **Setup Flight Modes:** The flight modes described in II-D1 were setup using the online guide [37]. The RC-channel 6 was used to toggle between three different flight modes. Before flight or during flight, these flight modes could be changed by using Mission Planner together with the

Telemetry Module. The base configuration were, state-1:Manual, state-2:FBWA, and state-3: various modes, depend on the mission.

- **Pre Flight Check:** Before flying the UAV, it needs to be armed. This features exits to prevent the motors from turning and to prevent takeoff before the pilot is fully configured and ready. To configure the arming process the online guide [38] was used. The arming was utilized in two steps, first arming the control surfaces and then arming the motors. Arming the control surfaces was done by pressing an arming button placed on the fuselage. To arm the motors the arming check, described in [39], first had to check a set of conditions. If one or more conditions fail to pass, the UAV will not arm. These conditions were changed to align with the set of hardware that was used for this UAV.

The conditions used:

- GPS connection
- Healthy sensors
- Connection with operator
- No throttle input from operator
- Neutral roll, pitch and yaw input from the operator
- **Tuning ESC:** BLheli32 ESC were used. They were configured with Arduino Nano and the software BLHeliSuite32. A guide on how to program any Arduino into a BLheli32 programmer can be found in [40]. A guide on how to program a BLHeli32 ESC with BLHeliSuite32 and an Arduino can be found in the following document [41].
- **Accelerometer and Gyroscope calibration:** Before using the accelerometer and gyroscope in the Pixhawk, they needed to be calibrated. This is a semi autonomous process done in Mission Planner. The operator will be asked to rotate the fuselage in various directions. A more detailed guide can be found at [42].
- **Compass Calibration:** Ardupilot needs to know how the compass is oriented about the fuselage. Compass calibration is a semi autonomous procedure done in Mission Planner. A complete guide can be found in [43].

IV. RESULT AND ANALYSIS

This section goes through the test flights performed during the thesis. These test flights should not be seen as the final result of the thesis, but rather as a part of the process of making a flying autonomous UAV. The idea with the flights was to start flying the UAV manually and then work towards completely autonomous flights, this is shown in Table II in column Flight Mode. Each flight test will be presented together with a flight analysis and the modifications done on the base platform, based on the knowledge gained from the flights. In Figure 16 the autonomous UAV Clouds can be seen before all the modifications and flight tests.

A. Flight 1

As shown in table II the first flight used two motors and flew in manual mode. There was no landing gear mounted, which led to the UAV being launched by hand.



Fig. 16. Picture of the autonomus UAV Clouds, built during the thesis

TABLE II
GENERAL DATA OF FLIGHTS

Flight Number	Date	Airfield/ Location	Motor Configuration	Flight Mode
1	2022-03-01	Gärdet	Dual	Manual
2	2022-03-11	Uppsala	Front	Manual
3	2022-03-30	Vallentuna	Front	FBWA
4	2022-04-06	Vallentuna	Dual	FBWA, Loiter
5	2022-04-06	Vallentuna	Daul	FBWA, Loiter
6	2022-04-12	Vallentuna	Dual	FBWA, Loiter, Auto

1) *Flight analysis:* The UAV was launched by hand and accelerated up to a speed of 16 m/s and a height of 30 m. When the pilot increased throttle, an undesirable roll occurred. The pilot slowed the UAV down and tried to land it, but the UAV was perceived as very difficult to fly and rear-heavy. During landing, the UAV crashed and this is shown in Figure 17.

2) *Modifications on the base platform:* Due to the undesirable roll, the two motors on the wings were removed and one was placed in the nose of the aircraft. This was done to get rid of the unsymmetrical thrust from the motors which had led to the undesirable roll. To solve the rear-heavy problem, more weight was added to the nose of the aircraft. To increase safety, landing gear was also fitted to avoid hand launching.



Fig. 17. Flight 1 Log-Data, Google Earth Pro



Fig. 18. Flight 2, UAV clouds flying in Manual

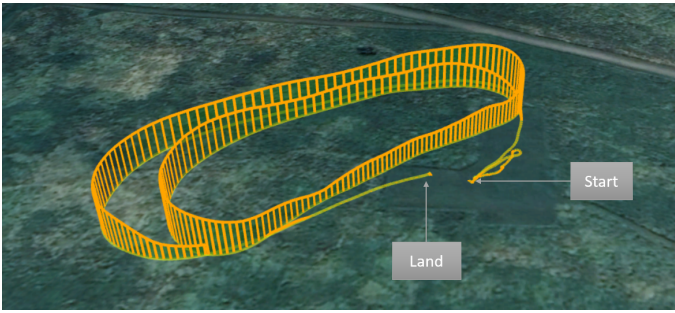


Fig. 19. Flight 2 Log-Data, Google Earth Pro

B. Flight 2

During flight 2, the UAV had one motor mounted on the nose of the aircraft and was equipped with the landing gear. The center of gravity was moved forward to the nose with an additional weight of 500 grams. The UAV was flown in manual mode and this is shown in Figure 18 during flight.

1) *Flight analysis:* The UAV accelerated on the ground getting up to a speed of 22 m/s. It was flown to an altitude of 42 m and then in two circles around the field before landing, which is shown in Figure 19. The pilot experienced that the UAV flew well but it was heavy and the RC transmitter was configured too aggressively.

2) *Modifications on the base platform:* After the flight, the control of the servos was configured to be less aggressive and new landing gear was installed to meet more difficult terrain during takeoff/landing.

C. Flight 3

The UAV was equipped with one motor at the front of the nose. The CG was placed 5 cm in front of the center of lift. The motor was slightly tilted downwards. The UAV was configured to fly in FBWA during the whole flight.

1) *Flight analysis:* During takeoff, the UAV started to yaw heavily to the left and it was difficult to make it fly straight during takeoff. But when the UAV got up to a speed of 10 m/s

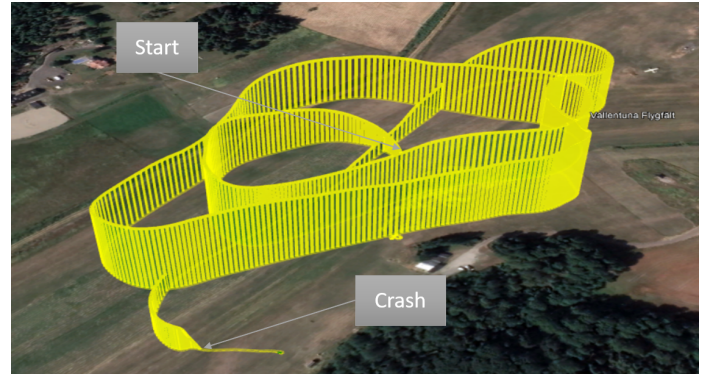


Fig. 20. Flight 3 Log-Data, Google Earth Pro

the flight properties became dominant, the UAV started to lift and became stable. At an altitude of 50 m, the UAV was flown in circles around the field. The flight ended unexpectedly in a crash due to one of the ailerons coming loose and the UAV immediately lost altitude. This is shown in Figure 20, where the altitude at the end of the flight drastically decreases. The autopilot tried to stabilize the pitch of the UAV, resulting in the UAV oscillating between having the nose horizontal to the ground and pointing straight down. This made the UAV lose speed and eventually crashed, with no big damage to the fuselage.

2) *Modifications on the base platform:* The aileron fell off because it was only held in place by thin foam, after repeated bending, by the servos, the foam had become weakened. All control surfaces were therefore reinforced with duct tape. After the flight, it was certain that the autopilot was capable of flying the UAV and was not the cause of the crash in the first flight explained in section IV-A. It was decided that it was time to install two motors instead of one onto the UAV to avoid the slipstream effect explained in section II-A7, causing the heavy yaw at the beginning of the flight.

D. Flight 4

The UAV was equipped with two motors, one on each wing. All the control surfaces were reinforced. Many taxing tests have been done on the ground and it was revealed that by incorporating differential thrust into the yaw-control, the UAV became much more controllable both in the air and on the ground.

1) *Flight analysis:* The UAV started with some difficulties while on the ground. When becoming airborne it was stable. The UAV flew to an altitude of 60 m and did some turns in FBWA mode, this is shown in pink in Figure 21. The UAV was switched to Loiter mode and turned 360 degrees, red in Figure 21. It was then switched back to FBWA mode, did some turns to lose altitude, and then land, orange in Figure 21.

2) *Modifications on the base platform:* During landing, the wheels of the UAV got stuck in the grass therefore the UAV hit the nose on the ground and turned upside down. One of the elevators broke in half as a result and was glued in place on site.



Fig. 21. Flight 4 Log-Data, Google Earth Pro



Fig. 22. Flight 5 Log-Data, Google Earth Pro

E. Flight 5

This flight was performed the same day as the flight in section IV-D, meaning the setup was the same.

1) *Flight analysis:* The UAV started in FBWA mode. It was flown to an altitude of 70 m and some turns were made, yellow in Figure 22. The UAV was then switched to Loiter and did one turn around the launch site, green in Figure 22. The UAV was put into FBWA mode and did some turns to lose altitude and speed to land, blue in Figure 22. During landing, the landing gear stuck into the grass and the UAV fell over once again. From Figures 23 and 24 it is possible to read the

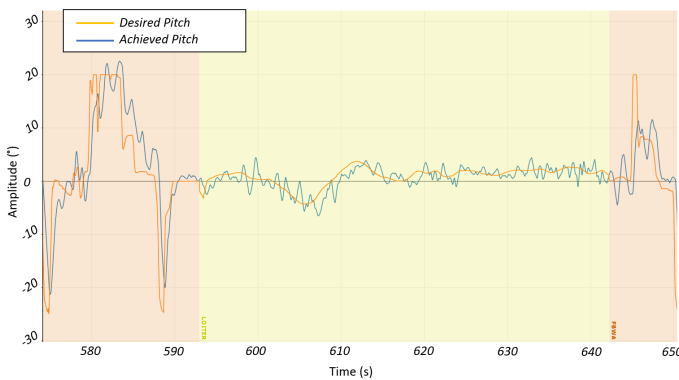


Fig. 23. Flight 5: Desired Pitch vs Achieved Pitch

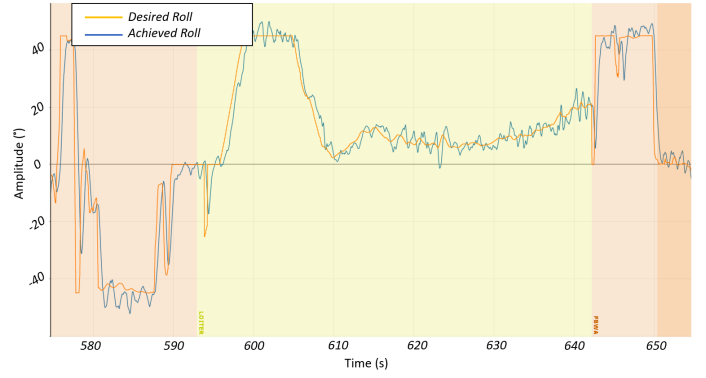


Fig. 24. Flight 5: Desired Roll vs Achieved Roll



Fig. 25. Flight 6 Log-Data, Google Earth Pro

desired and performed role/pitch. The lighter part of the graphs indicates flight in Loiter, where the curves converge with some oscillation. The same applies when flying in FBWA, the darker areas in the graphs, also here the desired role/pitch is carried out without major difficulties.

2) *Modifications on the base platform:* After Flight 5, the goal was reached, and the UAV could be flown in FBWA and autonomously without any problems. Some modifications were done to make the landing gear more stable. The pilot experience that the UAV was too stiff in roll and pitch, therefore the roll and pitch limits were decided to be increased.

F. Flight 6

The UAV had the same setup as the flights described in section IV-D and IV-E. It was flown with the purpose of data gathering and tuning the aircraft control system. The maximum desired roll angle was changed from 45 degrees to 60 degrees in the software. The loiter radius was changed from 60 meters to 40 meters. The maximum pitch angle was changed from 25 degrees to 40 degrees. This is to make the UAV more agile during turns and to take up less space when Loitering.

1) *Flight analysis:* During the start, the UAV was hard to get airborne. The snow had just melted and the landing gear got stuck in the ground because of the soft grass. When up to

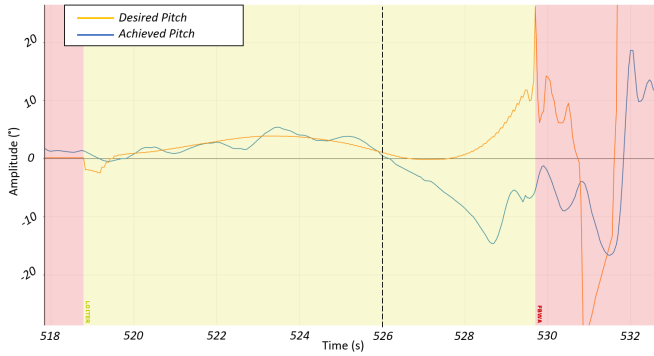


Fig. 26. Flight 6: Desired Pitch vs Achieved Pitch

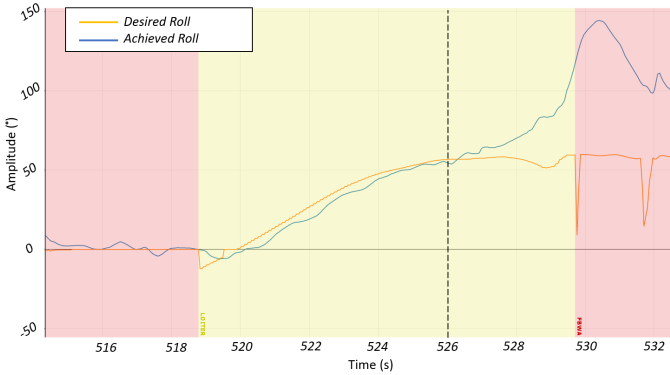


Fig. 27. Flight 6: Desired Roll vs Achieved Roll

a speed of 15 m/s, it flew well and the UAV was able to make sharp turns in FBWA. During the start, the UAV was flown in FBWA, green in Figure 25. It was switched to Loiter, blue in Figure 25, and started to do the desired turn but accelerated up to over 40 m/s and started to pitch down. The UAV was switched back to FBWA, but it had too much inertia to be able to control. It rolled to 60 degrees, rolled over, and crashed into the ground. In Figure 26 the undesired pitch can be seen from $t = 526s$, here the desired and achieved pitch started to diverge, which indicates that something went wrong. In Figure 27 the roll followed the same pattern, and the desired and achieved roll started to diverge.

2) *Modifications on the base platform:* After the crash, it was decided that it was not worth repairing the UAV. No mechanical conclusions were made from the final flight, but in future projects, the autopilot will be tuned with much greater care. The operators learned that after making changes they must be ready to switch to manual flight mode fast even after observing small deviation.

V. DISCUSSION

A. Landing Gear

The decision to use landing gear instead of hand launch did not only affect the start and landing procedure, but the whole testing phase of the aircraft. The landing gear made it possible to analyse the behavior of the aircraft during taxing tests, thus creating room for error without consequences being catastrophic. Between the flights described in IV, numerous

taxing tests were conducted with the purpose of testing and checking that the aircraft was stable enough for flight. The landing gear concept has been of such great importance that it will be used on the ALPHA UAV.

B. Differential Thrust

Using two motors resulted in many challenges due to uneven response with the same throttle signal. This was due to various factors. The motors were already heavily used and were worn unsymmetrical. The ESCs controlling the motors were programmed completely differently by default even though they were of the same model. These two factors lead to the motors giving different thrusts even though they received the same throttle signal. This problem was fixed by configuring both ESCs and by enabling differential thrust in Ardupilot.

C. Auto Tuning of the UAV

The crash explained in section IV-F could have been partially avoided if the Automatic-Tuning [44] flight mode had been used at an early stage of the flights. The changes that were made, especially the maximum roll angle, before Flight 6 were probably one of the reasons why the UAV went into a too sharp turn. Together with the unexplained acceleration, this may have led to the servomotors not having enough power to change the control surfaces seen in Figure 27 and in Figure 26, which eventually led to the crash. The Automatic-Tuning flight mode is a mode where the UAV gathers data while flying in FBWA, all the data gets used by the auto-tune code and results in new min and max values for the system. Instead of iterating through different min and max values, the Automatic-Tuning should be used from the start. This would probably have made the UAV fly even better, both in FBWA and during autonomous flights. Therefore this will be the first flight mode tested on the new UAV Nimbus explained in section V-F.

D. Center of Gravity

During the early test flights, the UAV was extremely unstable and tended to stall and turn unexpectedly. This was due to the center of gravity being too far back. To make the UAV stable and able to fly, the center of gravity needed to be in front of the center of lift. This was achieved by moving the battery as far into the nose as possible and by adding 700 grams of lead weights to the tip of the nose. These weights were later replaced by an additional battery.

E. Autopilot modes

In the beginning of the thesis, there were doubts regarding how well Ardupilot would handle the UAV, which led to greater emphasis being placed on trying to fly the UAV in manual mode. What we certainly can take with us from all flights is that the FBWA mode should be seen as the baseline mode before each flight. The problem with manual modes is the possibility to set the UAV in a stall if controlled incorrectly, where the FBWA mode has less chance of setting the UAV in dangerous control situations. The Auto-mode, explained in section II-D1, is the only flight mode which was not able to be tested because of the crash and needs to be further investigated during the upcoming UAV explained in section V-F.



Fig. 28. Crosswind Nimbus Pro 1900

F. Future Work

After the crash, explained in Flight 6, a new UAV was ordered with the name Nimbus and is shown in Figure 28. Nimbus is based on the same design as Clouds except the tail, but is of better quality, especially in terms of control surfaces and electronic connections. Nimbus will be constructed with the lessons taught by Clouds and will be of a much higher standard. Nimbus will be used to perform long endurance tests in auto mode and will eventually also test the Auto-Landing procedure embedded in Ardupilot. Nimbus will continue to serve as a test platform for the development of the hardware and software later to be used in the ALPHA project.

VI. CONCLUSION

The goal of this bachelor thesis was to implement an open source hardware/software platform and make it fly autonomously. The goals explained in section I-C served as the framework for the whole project and have also been met. Someone completely new to the field can build a sophisticated UAV out of readily available open source components. But it is important to be willing to try and expect failures on the way. The project is still in progress and hopefully, Nimbus will fulfill the goals more reliably.

ACKNOWLEDGMENT

The authors would like to thank our supervisor Mykola Ivchenko, he has supplied us with all the material needed to be able to learn and has always been positive. He gave our work purpose by aiming at the goal and giving us the bigger picture. The goal has always been to have fun while moving forward.

REFERENCES

- [1] N. Ivchenko, *ALPHA Project Handbook*, Stockholm, KTH, 2022.
- [2] Z. Firouzeh, M. Moradian, A. Safari-Hajat-Aghaei, and D. S. Mir-Mohammad-Sadeghi, H., "Design and implementation of ground station antennas for uav data radio link in uhf band," in *2006 2nd International Conference on Information Communication Technologies*, vol. 2, 2006, pp. 2195–2200.
- [3] S. Baldi, D. Sun, X. Xia, G. Zhou, and D. Liu, "Ardupilot-based adaptive autopilot: architecture and software-in-the-loop experiments," *IEEE Transactions on Aerospace and Electronic Systems*, Maui, HI, USA, pp. 1–1, 2022.
- [4] M. t. Benoît Henrivaux, "Integration of an open source flight controller into a fixed wing remotely piloted aircraft," n Liège, Wallonia, 2017.
- [5] I. Moir, *Aircraft systems : mechanical, electrical and avionics subsystems integration*, 3rd ed., ser. Aerospace series. Chichester: Wiley, West Sussex, 2008.
- [6] D. Howe, *Aircraft loading and structural layout*, ser. Aerospace series. Professional Engineering Publishing, London, 2004.
- [7] parallax. (2022, Apr) Steering an aircraft. [Online]. Available: <https://learn.parallax.com/tutorials/robot/elev-8/how-fly-multirotor-suav/steering-aircraft>
- [8] —. (2022, Apr.) Understanding rc propeller size. [Online]. Available: <https://www.rc-airplane-world.com/propeller-size.html>
- [9] M. t. R. Siddhesh, "The development of method (tool) to model propeller propulsion models to be implemented in the potential flow simulation software 'tornado', linköping," Ph.D. dissertation, Sep 2017.
- [10] —. (2022, Apr) Flight controllers explained for everyone.
- [11] R. Firoozian, *Servo Motors and Industrial Control Theory*, 2nd ed., ser. publisher. Berlin, 2014.
- [12] T. Yang, *Telemetry theory and methods in flight test*. Springer, Singapore, 2021.
- [13] —. (2021, Jan) Drönare pts. [Online]. Available: <https://www.pts.se/sv/bransch/radio/radiotillstand/ovriga-tillstand/dronare/>
- [14] R. Collinson, *Introduction to Avionics*, ser. Microwave and RF Techniques and Applications. Springer, Boston, 2012.
- [15] —. (2022, Apr) "motor-control considerations for electronic speed control in drones". [Online]. Available: <https://www.ti.com/lit/an/slyt692/slyt692.pdf?ts=1651243448241>
- [16] D.Kustec. (2022, Apr) Brushless motor kv rating explained. [Online]. Available: <https://dronenodes.com/brushless-motor-kv-rating-explained/>
- [17] J.Heydecke. (2022, Apr) Introduction to lithium polymer battery technology. [Online]. Available: https://www.jauch.com/downloadfile/5c5050fa5b6510e9a8ad76299baae4e53/white_paper_introduction_to_lipo_battery_technology_11-2018_en.pdf
- [18] —. (2022, Apr) Power monitor/module configuration in mission planner. [Online]. Available: <https://ardupilot.org/copter/docs/common-power-module-configuration-in-mission-planner.html>
- [19] —. (2022, Apr) Dc/dc switching regulators. [Online]. Available: <https://www.ti.com/power-management/non-isolated-dc-dc-switching-regulators/overview.html>
- [20] S. Östlund, *Eleffektsystem: EJ1200*. "KTH, Stockholm", 2007.
- [21] jimblom. (2022, Apr) Serial communication. [Online]. Available: <https://learn.sparkfun.com/tutorials/serial-communication/uarts>
- [22] A. Oudjida, M. Leam, and A. Ouchabane, "I2c-slave specification document, berlin," Dec 2005.
- [23] —. (2022, Apr) Flight modes. [Online]. Available: <https://ardupilot.org/plane/docs/flight-modes.html>
- [24] —. (2022, Apr.) Mission planner home. [Online]. Available: <https://ardupilot.org/planner/>
- [25] —. (2022, Apr) Downloading and analyzing data logs in mission planner. [Online]. Available: <https://ardupilot.org/copter/docs/common-downloading-and-analyzing-data-logs-in-mission-planner.html>
- [26] —. (2022, Apr) V-tail planes. [Online]. Available: <https://ardupilot.org/plane/docs/guide-vtail-plane.html>
- [27] —. (2022, Apr) Lm2595 simple switcher. [Online]. Available: https://www.ti.com/lit/ds/symlink/lm2595.pdf?ts=1651207681337&ref_url=https%253A%252F%252Fwww.google.com%252F
- [28] —. (2022, Apr) Pixhawk overview. [Online]. Available: <https://ardupilot.org/copter/docs/common-pixhawk-overview.html>
- [29] —. (2022, Apr) datasheet.m8n. [Online]. Available: https://content.u-blox.com/sites/default/files/NEO-M8-FW3_DataSheet_UBX-15031086.pdf
- [30] —. (2021, Apr) Configuring a telemetry radio using mission planner. [Online]. Available: <https://ardupilot.org/copter/docs/common-configuring-a-telemetry-radio-using-mission-planner.html>
- [31] —. (2022, Apr) Flashing frsky r9 modules. [Online]. Available: <https://www.expresslrs.org/1.0/quick-start/tx-r9m/>
- [32] —. (2022, Apr) Loading firmware. [Online]. Available: <https://ardupilot.org/plane/docs/common-loading-firmware-onto-pixhawk.html>
- [33] "-". (2022, Apr) Taranis x9d plus manual. [Online]. Available: <https://www.frsky-rc.com/wp-content/uploads/Downloads/Manual/X9DP/X9D%20PLUS-manual.pdf>
- [34] —. (2022, Apr) Radio control calibration. [Online]. Available: <https://ardupilot.org/plane/docs/common-radio-control-calibration.html>
- [35] —. (2022, Apr) Choosing servo functions. [Online]. Available: <https://ardupilot.org/plane/docs/servo-functions.html>
- [36] —. (2022, Apr) Plane failsafe function. [Online]. Available: <https://ardupilot.org/plane/docs/apms-failsafe-function.html>

- [37] —. (2022, Apr) Rc transmitter flight mode configuration. [Online]. Available: <https://ardupilot.org/plane/docs/common-rc-transmitter-flight-mode-configuration.html>
- [38] —. (2022, Apr) Arming plane. [Online]. Available: <https://ardupilot.org/plane/docs/arming-your-plane.html>
- [39] —. (2022, Apr) Pre-arm safety checks. [Online]. Available: <https://ardupilot.org/plane/docs/common-prearm-safety-checks.html#common-prearm-safety-checks>
- [40] —. (2022, Apr) Flash escs with any arduino! [Online]. Available: <https://www.flitetest.com/articles/flash-escs-with-any-arduino>
- [41] —. (2022, Apr) Use blheli suite32 with arduino. [Online]. Available: https://docs.google.com/document/d/1tqWfL0-P9IOzYpk1USsSwf4Q66_WxMP7Zx1CWyx19Ho/edit?usp=sharing
- [42] —. (2022, Apr) Accelerometer calibration. [Online]. Available: <https://ardupilot.org/plane/docs/common-accelerometer-calibration.html>
- [43] —. (2022, Apr) Compass calibration. [Online]. Available: <https://ardupilot.org/copter/docs/common-compass-calibration-in-mission-planner.html>
- [44] —. (2022, Apr) Automatic tuning with autotune. [Online]. Available: <https://ardupilot.org/plane/docs/automatic-tuning-with-autotune.html>

Obtaining Pitch Control for Unmanned Aerial Vehicle Through System Identification

Lucia Karens and Tawsiful Islam

Abstract—This study aimed to develop and evaluate a method to obtain a proportional-integral-derivative (PID) controller. The controller is for a control surface that controls pitch motion, by using data from flight tests with an unmanned aerial vehicle (UAV). Finding a suitable method to develop the controllers is essential to make the UAV autonomous, whilst being stable and controllable. Before developing the PID, data from test flights were used to model a transfer function for the control surface with MATLAB's toolbox for system identification. Thereafter, using the transfer function, the PID was developed by using MATLAB's toolbox for control systems. The whole method was evaluated by studying the rise time, settling time, and overshoot for the PID, and studying how well the transfer function fits with the flight data. The method of modeling the pitch motion with system identification and finding the PID gains has good potential to simplify the process of finding a PID controller. However, to acquire an accurate model for the pitch motion, which in turn can give a well-performing PID, an improved data sampling was suggested. Additionally, flight tests conducted before and after PID tuning, and in different conditions are recommended to be done in future studies. The flight test would work as a validation for the model to acquire a robust PID that performs as expected.

Sammanfattning—Syftet med denna studie var att utveckla och utvärdera en metod för att hitta en proportionerlig integrerande deriverande (PID) regulator. Regulatorn är för en kontrolllyta som kontrollerar tipp rörelsen genom att använda data från flygtester med en drönare. Att hitta en lämplig metod för att utveckla regulatorer är nödvändigt för att göra drönaren autonom, samtidigt som den är stabil och kontrollerbar. Innan PID:n utvecklades användes data från flygtester för att modellera överföringsfunktionen för kontrollytan med MATLAB:s programvara för systemidentifiering. Därefter, genom att använda överföringsfunktionen, utvecklades PID:n med MATLAB:s programvara för reglersystem. Hela metoden utvärderades genom att studera stigtid, insvängningstid och översläng för PID regulatorn, samt studera hur väl överföringsfunktionen modellerar flygdata. Metoden för att modellera tipp rörelsen och att hitta PID förstärkningarna har en god potential att förenkla processen av att hitta en PID regulator. Däremot för att få en precis modell för tipp rörelsen, vilket i sin tur kan ge en välpresterande PID, föreslås det att förbättra datainsamlingen. Dessutom rekommenderades det i framtida studier att flygtester genomförs i olika förhållande, både före och efter att PID regulatorn har hittats. Flygtesterna skulle fungera som en bekräftelse för modellen för att få en robust PID som presterar som väntat.

Index Terms—pitch control, flight tests, unmanned aerial vehicles (UAV), system identification, PID control

Supervisors: Mykola Ivchenko

TRITA number: TRITA-EECS-EX-2022:160

I. INTRODUCTION

Understanding the atmosphere of the Earth and the phenomena within it is fundamental to continuing to develop

telecommunications and other satellite-dependent technologies. Observations from the ground, satellites, sounding balloons, and sounding rockets are among the possible ways to observe and understand the atmosphere. In the last decades, unmanned aerial vehicles (UAVs) have been successfully used as observational platforms in a wide range of situations [1]. In order to observe upper-atmospheric phenomena such as aurora borealis, sprites, and blue jets, research in the Space and Plasma Physics Department and the Aeronautics and Vehicle Engineering Department of KTH Royal Institute of Technology has given rise to the student-driven project ALPHA [2]. The ALPHA project aims to develop a UAV that can fly above cloud level in order to take measurements and imaging of the high-altitude phenomena.

The ALPHA project is currently in its manufacturing phase. The half-scale UAV model is under construction, while the full-scale model is still in the modeling phase. Some parts of the UAV have been constructed already, and material tests are underway. The full-scale UAV will have a wingspan of around 4 meters and will be able to fly at around 10 km above sea level for several hours. [3]. A model of the ALPHA UAV is shown in Fig. 1.



Fig. 1. Current ALPHA design, courtesy of Victor Nan Fernandez-Ayala.

An important part of designing the ALPHA UAV is ensuring that it can fly autonomously. Part of the necessary work is to obtain an accurate model of the UAV. A model can be obtained through various methods, one of them is finding the model through an analytical method and taking help of simulations to find necessary parameters. However, previous studies have shown that simulations might not be suitable to model the true system without simplifications [4], [5]. An alternative method is modeling with system identification. System identification allows to model a system with observed input and output data

from empirical work. This method is a common method in the industry to model various dynamic systems and has the advantage of being quick, adaptable, and convenient [6]. This study thus uses system identification with an available UAV.

The aim of this study is to develop and evaluate a method to obtain a PID control system for the ALPHA UAV that controls pitch motion. This study uses system identification to model pitch ('nose up and down') motion by collecting and analyzing data from flight tests that are done with an available UAV.

II. THEORY: FUNDAMENTALS OF AERODYNAMICS

A. Basics of Flight

Aerodynamics is the study of the motion of air, and the forces and moments that apply to objects moving through the air. What makes an aircraft fly is a combination of forces and moments acting on the body and the wings of the aircraft. These forces are mainly gravity, thrust, lift, and drag. Designing an aircraft is to take into account these forces to ensure that the model can fly. Properties such as the mass of the aircraft, the surface area and the profile of the wings, the position of the center of gravity, and the shape and the surface area of the tail, all are important to make a viable aircraft [7]. The ALPHA team has spent many hours designing an aircraft that is stable for flight.

An aircraft can move in three dimensions. To describe the orientation of an aircraft, the terms *pitch*, *yaw*, and *roll* are used. In the following paragraph, airplane nomenclature is used. As the ALPHA UAV is an aircraft with fixed wings, the same nomenclature can be used for it.

Pitching is the rotation around the axis that goes through the center of gravity and is parallel to the wings. Yawing is the rotation around the axis that goes through the center of gravity and points vertically from the underside of the fuselage (the body of the airplane) to the top of the fuselage. Rolling is the rotation around the axis that goes through the center of gravity, from the tail to the nose of the airplane. In order to control these movements, the airplane has control surfaces on the wings and the tail. These surfaces can be deflected to induce one or several of the three rotations. To put it simply, on a conventional tail, there are two horizontal control surfaces called the elevators. Deflecting the elevators causes the airplane to pitch. There is also a vertical control surface on the tail, called the rudder. The rudder serves to induce a yawing motion. The control surfaces on the wings have a different name depending on where they are situated. If there is only one control surface on each wing, it is called the aileron. Deflecting the ailerons causes a rolling motion. The control surfaces and the rolling, pitching, and yawing axes are shown in Fig. 2.

The pitching motion is called longitudinal, and the yawing and rolling motions are directional and lateral. It is easier to study the longitudinal motion of the airplane because it can be decoupled from the lateral and directional motions [7], [8]. This means that pitching motion can be controlled entirely by deflecting the elevators while rolling and yawing motions are connected to each other and can not be easily studied independently of one another. This study has chosen to focus

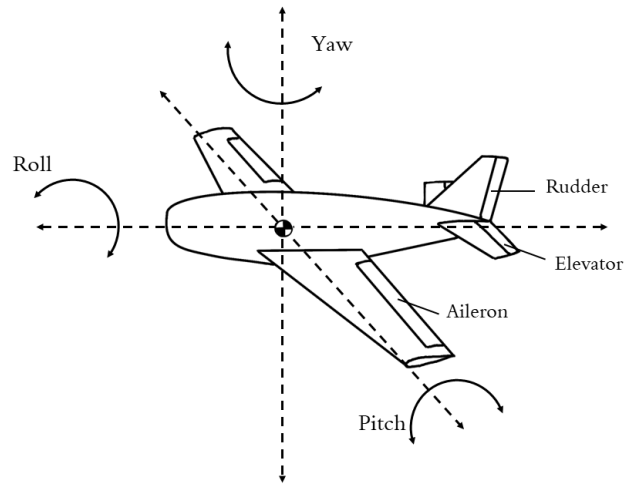


Fig. 2. Pitching, yawing and rolling axes for the aircraft.

on longitudinal control. Longitudinal control is control of the pitch, and of the pitch rate, which is the speed at which the pitch changes.

B. Reference Frames

Several different coordinate systems are used in aerodynamics. It is important to use the right frame of reference when looking at the system responses of the aircraft.

1) *The Body Fixed Frame*: The frame is orthogonal. The origin point of the frame of reference is the center of gravity of the aircraft. The x-axis points forward through the nose of the aircraft and is also the roll axis of the aircraft. The z-axis points perpendicularly down and is the yaw axis. The y axis (pitch axis) is perpendicular to the xz-plane and points to the right in accordance with the right-hand rule.

2) *The North East Down (NED) Frame*: The frame is orthogonal. Its origin is a point on the surface of the Earth or the center of gravity of the aircraft. The x-axis follows the vector line that points to the magnetic North (tangential to the meridian). The z-axis points down towards the center of the Earth. The y-axis points East, tangential to the parallel. The frame is shown in Fig. 3.

The Body Fixed frame does not give information about the orientation or the position of the aircraft. To take these into account, a change of frame of reference is needed, from the Body Fixed frame to an external fixed frame, for example, the NED frame with origin at the center of gravity of the aircraft. Conversion from one frame of reference to the other is done by three consecutive rotations of the reference frame, with so-called Euler Angles [7].

The frame of reference used in this paper is mostly the NED frame since it is the frame used for obtaining equations of motion of the aircraft, and the relevant frame for collecting data.

C. Equations of Motion

The equations of motion of an aircraft are obtained by taking into account all aerodynamic parameters and forces

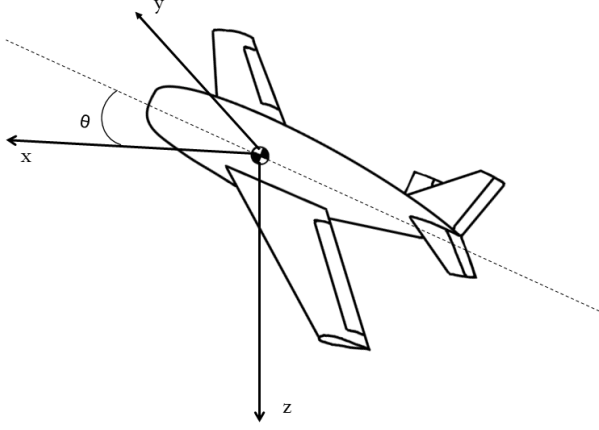


Fig. 3. North East Down frame for the aircraft. In this figure, the aircraft's nose is facing North and is pitching up at angle θ .

acting on the aircraft. By using Newton's second law of motion and switching from the Body Fixed Frame to the NED reference frame, the equations of motion can be obtained [7]. The following Laplace transformed equations are linearized using the small-disturbance theory. This theory assumes that the aircraft only experiences small deviations around a steady flight condition. The equations are also somewhat simplified by assuming that controls are held fixed and that the propulsion is constant. Even though they are simplified, the equations give a good general description of the system [7]. The linearized and simplified equations of longitudinal motion for an airplane are of the form

$$\frac{\Delta q(s)}{\Delta \delta_e(s)} = \frac{As + B}{s^2 + Cs + D} \quad (1)$$

and

$$\frac{\Delta \theta(s)}{\Delta \delta_e(s)} = \frac{As + B}{s^3 + Cs^2 + Ds}, \quad (2)$$

where $\Delta q(s)$ is the change in pitch rate, $\Delta \theta(s)$ is the change in pitch, $\Delta \delta_e(s)$ is the deflection angle of the elevators. The coefficients A, B, C, D depend on the aerodynamic parameters of the aircraft and on its physical properties. For instance, the mass of the airplane, the position of its center of gravity, the velocity, the lift and drag forces, the aerodynamic moments, the current position of the elevators, the current pitch of the aircraft, among others, are all contributing to the coefficients. Detailed equations are presented in [7]. Note that to switch from pitch rate to pitch, a simple multiplication by the integration factor $1/s$ is done. These equations are for so-called short-period motions, which are motions when a small deviation from equilibrium occurs.

III. THEORY: AIRCRAFT

As the ALPHA UAV is still in the modeling phase, and the half-scale ALPHA UAV is in the assembling phase, they are not ready for flight testing. The flight testing is thus done with another radio-controlled aircraft, X-UAV Clouds [9]. This aircraft is commercially available and is marketed towards and used by amateur RC plane pilots. This plane was

chosen for its size, which is close to the half-scale ALPHA UAV, and its availability. When it was made clear that the half-scale ALPHA UAV would not be manufactured in time for context L4a and L4b to perform necessary flight tests and other studies, group L4a and the ALPHA flying team acquired Clouds. Clouds was already in possession of the KTH Aeronautics Department. This study is based on data and analysis of Clouds.

Clouds is a fixed-wing plane made of EPO foam, with a V-tail and 1880 mm wingspan. The controllable surfaces of Clouds are the ailerons on the wings and the ruddervators on the tail. There are two propellers on the aircraft, each driven by a motor. The aircraft uses differential thrust, which means that the motors can act independently of one another.

Clouds has been outfitted with a custom electronic system by group L4a [10]. They have also, together with the flying team in ALPHA, made several modifications, including but not limited to, adding landing gear, adding a 3D-printed nose, and a belly plate to make the aircraft more resilient to crashes. An adhesive has also been used to ensure that every part of the aircraft is put in place, and to make the control surfaces more sturdy. The aircraft's appearance in April 2022 can be seen in Fig. 4.



Fig. 4. X-UAV Clouds on April 6th, 2022.

Clouds is a V-tail plane, while the ALPHA UAV has a more conventional T-tail. A V-tail and a conventional tail do not have the same control surfaces. While a conventional tail (and a T-tail) typically has a rudder and two elevators, the V-tail has two ruddervators, which serve as both rudder and elevator. One thing that should be noted when working with aircraft with V-tails is that V-tail dynamics are more complex than conventional tail dynamics. Fortunately, ruddervators can be approximated as elevators if the control surfaces are deflected at the same time and in the same direction [4]. This consequently allows one to use the equation of motion described in section II-C. Another argument for approximating the V-tail to a conventional tail is that the method is developed with the ALPHA UAV in mind, which has elevators and a rudder on its tail.

The hardware and sensors used in Clouds according to [10] result in that the ruddervators and other servos in Clouds do not give feedback to the onboard controller. They act as "black

boxes”. This means that it is not possible to obtain data on the actual deflection angles of the ruddervators during flight. It is, however, possible to obtain data for the desired deflection angles sent by the onboard controller to the ruddervators. The ruddervators can be modeled by a simple transfer function [7]:

$$\frac{\delta_e}{\delta_c} = \frac{k}{\tau s + 1} \quad (3)$$

where δ_e is the actual deflection of the ruddervator acting as an elevator, δ_c is the deflection sent by the controller, and k and τ are parameters of the servo and its motor. By taking into account this transfer function, the transfer function for the whole servo-dynamic equations of motion system can be modeled. The transfer function for the whole system is the product of (1) and (3) for the pitch rate, and of (2) and (3) for the pitch. The transfer functions of the pitch rate and pitch depending on the angle sent by the controller to the ruddervator are of the form

$$\frac{\Delta q}{\Delta \delta_c} = \frac{A's + B'}{s^3 + C's^2 + D's + E'} \quad (4)$$

and

$$\frac{\Delta \theta}{\Delta \delta_c} = \frac{A's + B'}{s^4 + C's^3 + D's^2 + E's}, \quad (5)$$

where A', B', C', D', E' are coefficients resulting from the multiplications. The complete transfer function thus depends on both the aerodynamic properties of the aircraft and the properties of the ruddervator. The servo response is thus part of the whole system's response. Servos with a faster response time make the entire system respond faster.

IV. THEORY: FLIGHT CONTROL

A. Obtaining the Transfer Function

To design a proportional–integral–derivative (PID) controller, the system that is studied must first be understood and made explicit. For an aircraft, this means obtaining the transfer functions for the servos and the transfer functions that depend on the aircraft's aerodynamics. There are several ways of obtaining the relevant transfer functions of the system: by simulating, using wind tunnels, and conducting test flights. Each method has its advantages and drawbacks.

Simulating an aircraft makes it possible to obtain aerodynamic parameters without having to fly the aircraft. This is usually done during aircraft design, to ensure that it is stable and flies well. A model of the aircraft is made using Computer-Aided Design (CAD) and then put through Computational Fluid Dynamics (CFD) software to obtain necessary aerodynamic parameters. To be accurate, these simulations use very fine 3D meshes and do heavy calculations. Simulating is often a very time-costly effort.

To use wind tunnels, the aircraft needs to be manufactured. It can be a full-scale aircraft or a smaller model. The wind tunnel then gives data about the aerodynamic properties of the aircraft, without needing to fly the aircraft. Wind tunnels are not easily accessible and are expensive and time-consuming. They do not give very accurate data for smaller UAVs, because of the low speeds, the non-traditional characteristics of the

UAVs (lighter mass, unconventional geometry, and more), and the limitations regarding stall, among others [5].

Test flights are relatively easy to conduct and give realistic data. They are the best way to get the real parameters. But they need the aircraft to be fully constructed and designed, and they need time and a pilot. The data obtained might also be inexact due to noise from measuring instruments and atmospheric conditions. For UAVs, especially small ones, doing flight tests is considered to be an advantageous way to obtain the needed parameters [5], [11].

Since Clouds is already manufactured and can be used for test flights, it was decided to use the test flight method to obtain the needed parameters.

B. Modeling with System Identification

Modeling the pitch motion with data from flight tests can be done with system identification. System identification is a tool to build a mathematical model for a dynamic system. For this study specifically, that would be the pitch motion that is controlled by the ruddervator. In the industry, modeling dynamic behaviors with system identification is an established method. This is mostly thanks to available software like MATLAB system identification toolbox, making it time-efficient compared with analytical methods. System identification is done by using observed data for the system from empirical tests to estimate the model through algorithms for identification. The computer goes through iterations of estimating the model by trying to fit the model to the data and validating it before presenting it to the user. The user's role is to filter data that may be unreliable or remove noise that may lead to an inaccurate model before letting the computer estimate a model. Moreover, the user also chooses the model structure, for instance how many poles and zeros the transfer function should have, or whether the model should be nonlinear or a state-space model. After the estimated model has been presented to the user, they have to do further analysis before accepting it [6], [12].

There are several decisions that need to be considered by the user when using system identification to estimate a model. The first is deciding which input and output signals to measure, to make sure they can describe the system that will be modeled. In addition, the user needs to consider if the signals are measured from an open-loop or closed-loop system. Data from an open-loop system have the advantage that the input is independent of the output. It is also easier to estimate and analyze the model if the observed data are from an open-loop system. There are cases where observing data from a closed-loop system might be needed due to the system being unstable if the output is not controlled. However, this should be avoided if possible as the model may be inaccurate due to the dependent signals [6], [12].

The second essential decision that should be made is the sampling of data, how many data points should be collected, and what the sampling rate should be. Both too fast and too slow sampling can be inconvenient. Too fast sampling leads to redundant data where new data does not provide further information. This is of interest if there is limited data storage space. Too slow sampling is in general worse than

too fast sampling. Having few data points makes it difficult to determine the parameters for the model, which is why it should be avoided to determine a reliable model [6], [12].

C. PID Controller and Performance Specifications

In order to have a stable system, the output from the system needs to be controlled, which can be done by using a PID controller. As the PID is a simple controller, it is easy to implement the PID when it uses the reference and output signal to calculate an error. The error is then used to control the input until the output matches the desired reference signal. On account of their simplicity, PID controllers are implemented in various circumstances, software, and even in modern-day aircraft. A block diagram of the closed-loop system studied in this paper is shown in Fig. 5. The open-loop system includes only the servo and the pitch dynamics.

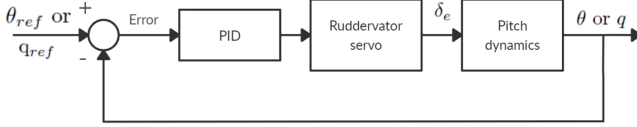


Fig. 5. Closed-loop block diagram for pitch and pitch rate.

The time dependent output signal $u(t)$ is controlled according to the following equation:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de}{dt} \quad (6)$$

where $e(t)$ is the error signal and is the difference between the reference signal and the output signal. The constants K_p , K_i , and K_d are the gains for the proportional, integral, and derivative controllers in the PID. Obtaining these gains can be done with different analytical methods by working with root locus or using the Ziegler-Nichols method for instance [7]. However, analytical methods can be time-consuming and may have their drawbacks. In certain cases, it is advantageous to reduce a higher-order system to a second-order system, which makes calculations easier. However, the complexity of the system is not fully accounted for when designing the PID and the airplane might respond differently from what is expected. Other ways to design a PID can be done with software such as MATLAB. In MATLAB an application from the extension Control system toolbox [13] can be used by tuning a response from a transfer function and acquiring the gains from the tuned response numerically. With the application, the response from a higher-order system can be tuned directly, which allows the user to acquire gains for the PID that are more accurate than ones obtained analytically when reducing higher-order systems [14].

There are two different ways to analyze a controller's performance, either by doing it in the time domain or the frequency domain. Within the frame of this paper, the interest in performance lies in how fast the system responds rather than the cyclic behavior of the system. Therefore a time-domain analysis is used in this study. Time-domain specifications

one can study are rise time t_r , settling time t_s , and peak overshoot M_p . The rise time of a response describes how fast the output changes from 10% to 90% of the desired output. The settling time is the time it takes for the system's output to stay within an interval of $\pm 5\%$ of the desired output. Maximum overshoot is given in percentage and is the size of the maximum output relative to the desired output. How to study the time specifications is shown in Fig. 6 [7].

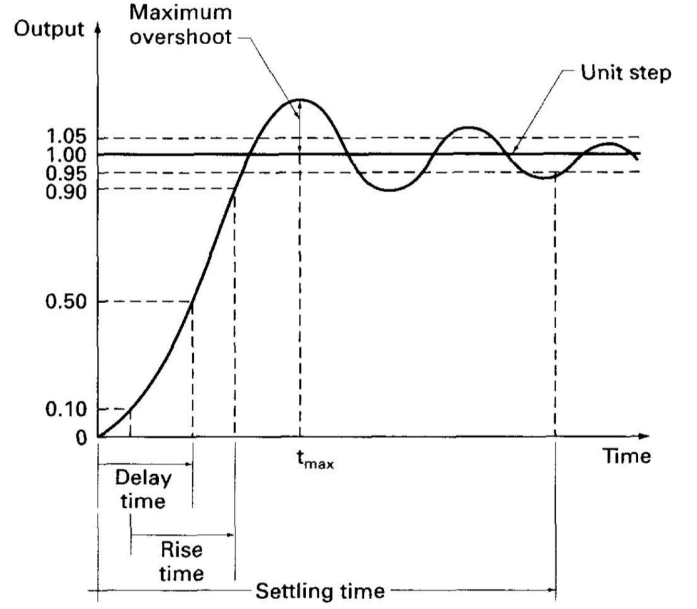


Fig. 6. Time response of an output showing how the time specifications are identified. Figure is retrieved from [7].

When designing a controller in general, it is preferred to have t_r , t_s , and M_p as small as possible. Sometimes this may not be possible in practice due to the physical limitations of the system. In these cases, design choices are made such as tolerating a maximum of 5% overshoot or allowing the rise time or the settling time to be within a time interval.

V. THEORY: FLIGHT TESTING

A. Flight Test and Flight Logs

Flight testing is an integral part of determining whether an aircraft is fit for flight or not. Flight testing is conducted after the aircraft has been designed and gone through CAD and CFD simulations to ensure its static stability. Flight tests are also an alternative to obtaining data on how the aircraft responds to phenomena that are difficult to simulate, or non-linear. This is the case when wanting to find the transfer functions for the aircraft. The functions obtained from flight tests may look different from the functions with calculated aerodynamic parameters from simulations as simulations exclude or simplify non-linear phenomena.

When conducting flight tests, a proper pre-flight checklist and flight logging are needed to ensure that flight tests are streamlined and that relevant data are obtained and stored correctly. Flight logging is also necessary to document technical issues and possible crashes. Establishing flight protocols

increases safety and improves the efficiency of conducting flight tests [15], [16].

B. Autopilot

Contexts L4a and L4b discussed the benefits and disadvantages of creating an autopilot from scratch and using an already available autopilot. Creating an autopilot complex enough to control the entirety of the aircraft, conduct pre-planned missions, and fly the aircraft safely, is an enormous task that is beyond the scope of our project. Thus it was decided to use an already available autopilot called ArduPilot.

ArduPilot is free, open-source software that enables users to control and use UAVs, such as helicopters, planes, and others [17]. ArduPilot is widely used by both amateur UAV pilots and students, its versatility making it a valuable asset for piloting UAVs. The accessibility, relative simplicity, and versatility of the software make it ideal to use for the ALPHA project and is the reason why context L4a and L4b decided to use ArduPilot for piloting the Clouds prototype.

ArduPilot connects the aircraft with a ground control station (GCS), most often a computer. The aircraft itself carries sensors, output devices such as motors and servos, and a controller, which is a small computer-like device that takes inputs from the sensors and sends outputs to the output devices. The controller is run by a code that can be downloaded from ArduPilot to fit the specific type of UAV. The GCS is the interface between the user and the controller. ArduPilot's software is Mission Planner [18], which is a program that can be downloaded to any personal computer. Mission Planner makes it possible for the user to control their UAV, download and analyze output data, create detailed missions, and more.

One of the many tools available through Mission Planner is setting flight modes for the UAV. The flight modes that are included in Mission Planner come with presets for the settings on how the UAV can fly. One of the modes that allow the UAV to have assisted flying is called fly-by-wire A (FBWA). FBWA assists the pilot to fly the UAV by limiting how much the aircraft can roll and pitch. However, the elevators are still manually controlled. The maximum and minimum angles (in degrees) that the aircraft can roll and pitch are set by the pilot in Mission Planner. The throttle is manually controlled by the pilot when flying with FBWA [19].

VI. METHOD

Designing a controller for the pitch of the aircraft was done in several steps: Obtaining the open-loop response of the system from flight tests, modeling the open-loop response with system identification and finding the PID gains that improved and stabilized the closed-loop response for the pitch and pitch rate. The method is implemented on the pitch angle and pitch rate separately from each other. The evaluation of the method was also done for the separate cases.

A. Flight Testing

Flight testing has been done mainly by the ALPHA flying team, coordinated and led by Augustsson and Barsby by

using X-UAV Clouds with the configuration and specifications according to their report. Before flying to collect data for the pitch or from other servo tests, flight tests were done to ensure that Clouds was trimmed and other parameters were set correctly. This is a necessary step to check that the aircraft was capable of flight, take-off, and landing in order to reduce the risk of crashes [10].

Thereafter, the test for pitching the aircraft with the ruddervators was performed. The test was done in FBWA flight mode to have manual control over the ruddervators without any interference from built-in controllers from Ardupilot. FBWA was also chosen to minimize the risk of stalling the aircraft and crashing, which can be caused by low airspeed when pitching too much [19]. The aircraft flew a straight line with no input for roll or yaw by the pilot. The initial speed before the test was approximately 15 m/s and no throttle inputs were made by the pilot during the test. The pitching was performed for ten seconds where the pilot pitched up and then pitched down. This was repeated immediately after and each pitching motion was executed for about 2-3 seconds.

A flight logging protocol was written, and a pre-flight checklist was created in collaboration with group L4a. The flight logging template is given in the Appendix. The tests that have been performed have been logged in flight logs and the data were saved in tlog files. The tlog files can be further analyzed; data from the tlog files were used to model the response from the ruddervators when changing the pitch.

B. Determining Transfer Function from Open-loop Response

System identification was used by importing flight data from the test flights into an application for system identification that is included in the MATLAB system identification toolbox. The input was the signal sent from the remote control to the servos, and the output was the pitch of the aircraft. The tlog file that was stored after the flight test contained three relevant data sets for system identification. The data set named `chan2_raw_mavlink_rc_channels_t` contained the input signal from the remote controller, and the output signals were named `pitch_mavlink_attitude_t` for pitch and `pitchspeed_mavlink_attitude_t` for pitch rate. The input was a pulse-width modulation (PWM) signal, which controls the servos with electrical signals. The reason why PWM signals were used as input data was to account for the response time from the servos that affected the total response time. PWM signals are proportional to deflection angles and can be converted if the corresponding PWM signals for maximum deflection angles are known. However, with Clouds and tlog files, exact and true deflection angles are difficult to determine and therefore had not been chosen as the input signal in this study. The output data were in degrees and degrees per second. The data for the pitch changes were imported to the System Identification application. The application required that the two data sets have the same amount of data points and the same sample time. However, due to Cloud's hardware's setting on sampling data, the number of data points did not match and the signals were sometimes sampled at different rates. Therefore, simplifications and assumptions had to be made in order to

determine the transfer function. The number of data points for the pitch angle and pitch rate was almost double compared to the data points for the PWM signals over the same time interval. To match the number of data points, every second data point for the pitch was removed. Thereafter the sample time was assumed to be the number of data points over the time interval. Besides input data, output data, and sample time, no further changes were done in the settings for the system identification.

In the application, a transfer function could be estimated from the data by setting how many poles and zeros the transfer function would have. For the open-loop response, a second, third and fourth-order transfer function with none to four zeros were tested to see which of them fitted the flight data the best. The fit percentage was calculated by

$$fit = 100 * (1 - \frac{\|y - \hat{y}\|}{\|y - \bar{y}\|}) \quad (7)$$

where y was the original output from the flight data, \hat{y} was the output calculated from the estimated transfer function, and \bar{y} was the mean of the original output. The norm of the differences was calculated in the formula. The application gave a number in percentage of how well the model matched the flight data [20]. The fit is thus how well the model matches with the data.

C. Designing a PID Controller

After a transfer function for the ruddervators had been estimated, a PID could be determined; to stabilize the system, or possibly make it respond faster. A controller for pitch and a controller for pitch rate were created separately. The pitch PID controller had the error of the pitch as input and the pitch rate PID controller had the error of the pitch rate as input.

The controller was selected to be a parallel PID to find the gains in (6) and was also chosen to be a one-degree-of-freedom PID in the PID tuner application. The gains for the PID were determined by using the PID tuner application, which is included in MATLAB control system toolbox. The response could be tuned by adjusting two sliders in the application. One slider adjusted the response time of the output and the second adjusted the transient behavior of the response, whether it should be robust or not. The criteria that were followed when designing the PID were inspired by the specifications Onuora et al. used [21]. The specifications that the PID needed to fulfill were prioritized in the following order that the:

- 1) overshoot M would be less than 10%;
- 2) rise time t_r would be less than 0.2 seconds;
- 3) settling time t_s would be less than 0.5 seconds.

In order to consider physical limitations of how fast the aircraft could pitch, the times were also chosen to be close to the desired time limit, even though a faster time response could have been chosen. Behaviors like oscillation in the response were minimized as much as possible by first making the response more robust. If that would not work, the response time was increased.

VII. RESULTS

Several flight tests were conducted. Unfortunately, due to the many setbacks in making Clouds fly [10] and the large number of preliminary tests that were necessary for general flight; pitch collection could only be performed once in the desired conditions, at the beginning of April. During the following flight session, Clouds crashed irreparably.

From the successful flight test, the isolated pitch changes were performed for 10 seconds. From the flight test, 20 data points were obtained for the PWM input, and 40 data points were recorded for the pitch angle and pitch rate. In order to use the System Identification application, the pitch and pitch rate were undersampled by removing every second data point starting from the second data point. Doing this gave a sample time of 0.5 seconds. The comparison between the original data set and undersampled data set is shown in Fig. 7 for pitch angle and Fig. 8 for pitch rate. Both figures also show the PWM signals for the pitch motions during the flight test.

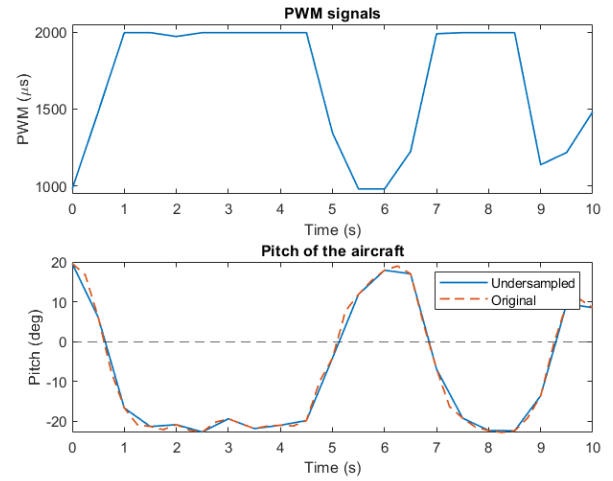


Fig. 7. Input PWM signals and output pitch angles for the aircraft during the flight test.

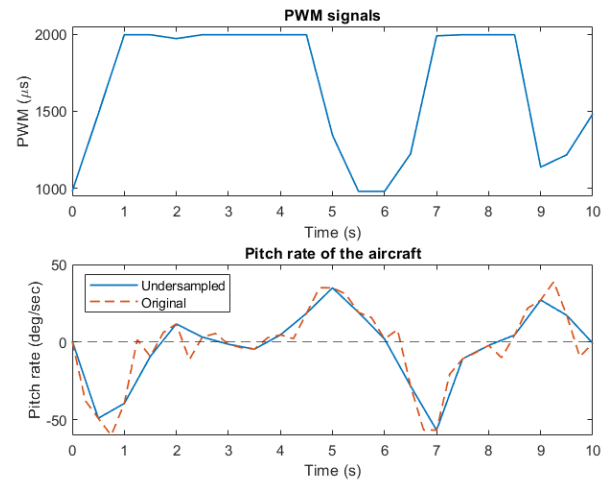


Fig. 8. Input PWM signals and output pitch rate for the aircraft during the flight test.

A. PID for Pitch

After using system identification to model the pitch, different numbers of poles and zeros were tested to see if they fit the data from the flight test. Fig. 9 shows the tested models where the first number in the name stands for the number of poles and the second stands for the number of zeros. The graphs in the figure show the unit step response for the transfer functions that were approximated for the pitch θ . The two transfer functions that had the best fit compared to the data from the flight test were the following:

$$\theta_1(s) = \frac{-9.83 \cdot 10^{-5}s^4 - 0.059s^3 - 0.00455s^2 - 0.175s - 0.02}{s^4 + 2.03s^3 + 4.49s^2 + 5.9s + 4.43} \quad (8)$$

and

$$\theta_2(s) = \frac{-0.0108s^3 + 2.77 \cdot 10^5s^2 - 7.12 \cdot 10^5s - 6.55 \cdot 10^4}{s^3 + 4.39 \cdot 10^6s^2 + 2.10 \cdot 10^7s + 1.26 \cdot 10^7}, \quad (9)$$

where θ_1 , with four zeros and four poles, had a fit of 80.6% and root mean square error of 3.01 degrees, and θ_2 , with three zeros and three poles, had a fit of 77.1% and root mean squared error of 3.56 degrees.

For the PID, θ_1 was used to tune the response for the aircraft's pitch. The gains and the time specifications that were achieved with the PID tuner for the pitch and the unit step response are shown in Table I and Fig. 10.

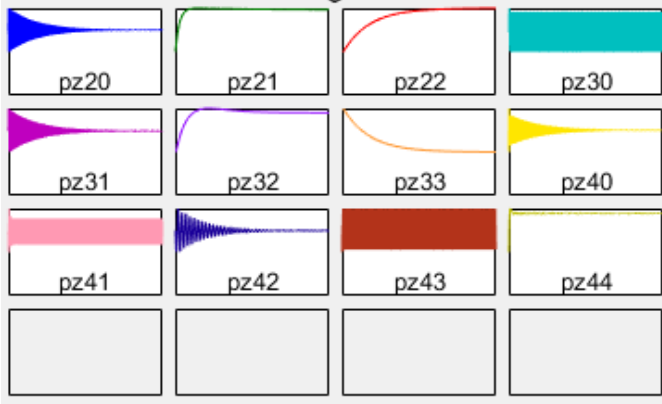


Fig. 9. Tested transfer function models and the shape of their unit step responses are shown in the screenshot from the System identification application. The number of poles is the first number and the number of zeros is the second number in the name of the transfer function. The y-axis shows the amplitude of the response and x-axis is the time. The scale between the graphs varies.

TABLE I
GAIN VALUES AND TIME SPECIFICATION FOR PITCH'S PID

Parameter	Value
K_p	-2581
K_i	-31 963 316
K_d	0
t_r	0.00058 s
t_s	0.00419 s
M_p	9.79%

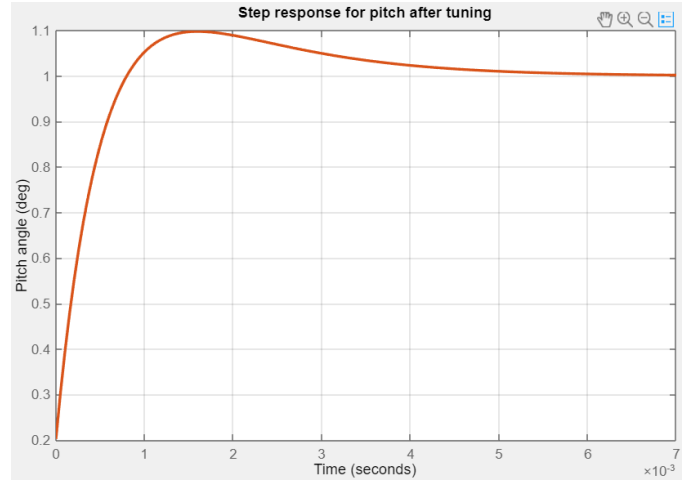


Fig. 10. Unit step response for the pitch after PID tuning.

B. PID for Pitch Rate

Fig. 11 shows the tested models for the pitch rate. The graph for each transfer function also shows the unit step response. The two transfer functions that had the best fit for the pitch rate q when compared to the data from the flight test were the following:

$$q_1(s) = \frac{-0.0475s^3 - 0.0289s^2 - 0.303s - 0.00421}{s^3 + 2.32s^2 + 5.51s + 12.8} \quad (10)$$

and

$$q_2(s) = \frac{-0.0226s^4 - 0.0367s^3 - 0.0465s^2 - 0.0727s - 0.00235}{s^4 + 0.698s^3 + 6.51s^2 + 0.976s + 7.14}, \quad (11)$$

where q_1 , with three zeros and three poles, had a fit of 68.72% and root mean square error of 7.21 deg/sec. Transfer function q_2 , with four zeros and four poles, had a fit of 75.87% and root mean square error of 5.57 deg/sec.

For the PID, q_1 was used to tune the response for the aircraft's pitch rate. The gains and the time specifications that were achieved with the PID tuner for the pitch rate and the unit step response are shown in Table II and Fig. 12.

TABLE II
GAIN VALUES AND TIME SPECIFICATIONS FOR PITCH RATE'S PID

Parameter	Value
K_p	0
K_i	-1363
K_d	0
t_r	0.0371 s
t_s	0.314 s
M_p	0%

VIII. DISCUSSION AND ANALYSIS

A. Flight Tests

Flight tests were more difficult to perform than anticipated. Hardware problems such as trimming failures; motor synchronization problems; difficulties in properly connecting all

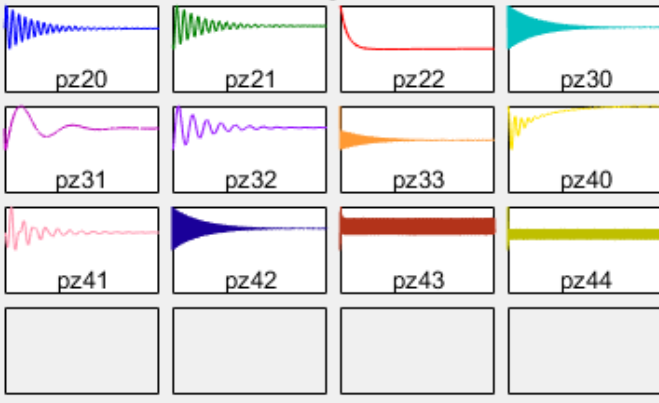


Fig. 11. Tested transfer function models and the shape of their unit step responses are shown in the screenshot from the System identification application. The number of poles is the first number and the number of zeros is the second number in the name for the transfer function. The y-axis shows the amplitude of the response and x-axis is the time. The scale between the graphs varies.

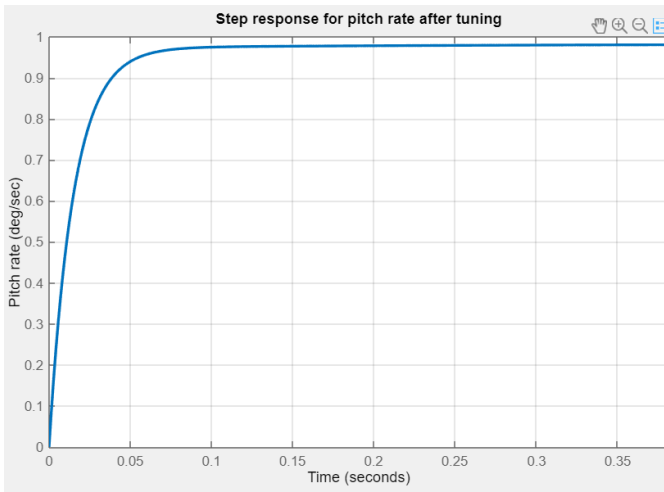


Fig. 12. Unit step response for the pitch rate after PID tuning.

electronic systems with each other and with Mission Planner; inappropriate propeller sizes; and others [10], led to a limited number of flight tests in the time of this study. This was outside of this group's control. In addition, the flight tests where Clouds managed to take off were divided into flights where the flying team learned how to pilot Clouds, flights that tested various flight modes from Ardupilot, and flight tests for collecting data on pitching, which were the flight tests of interest for this study. Pitching tests were not fully prioritized, which led to the result that only one test was performed before the crash. In addition, the flight tests were performed for the most part in FBWA mode, which is not entirely manual, because context L4a and the flying team wanted to minimize the risk of crashes as much as possible. For our study, it would have been better to have results from manual mode, where the pilot has full control of the aircraft, to get a fully accurate model of the system. Flying with FBWA was a compromise between flying fully manual and flying with aid from Ardupilot. As discussed earlier, FBWA does

not correct pitching angles but limits them.

The crash led to an abrupt end of the flight tests. The crash took place during the testing of one of the automatic flight modes of Ardupilot. The possible causes of the crash are explored more in detail in [10], but it seems to have been a mixture of flying too low, setting a more aggressive flying style than should have been used, and not setting a maximum speed for Clouds. The crash showed that while Ardupilot is an advanced tool that has many useful options, it does not prevent crashes from happening, and might even cause them if not used correctly. The crash was proof that in order to conduct safe flight tests, it is very important to understand Ardupilot's parameters and functions in great detail. Other conclusions are that flight tests which are for determining the core system's responses (without PIDs or other limitations or help from Ardupilot) should be performed before testing the autopilot functions. This might seem counter-intuitive since the aim of ALPHA is for the UAV to be autonomous. But understanding the system response first is necessary to develop a good PID that will then be a part of the autopilot.

B. PID for Pitch

Regarding the data for the pitch, 20 data points were used in the end for modeling the transfer function for the pitch motion from the ruddervators, which resulted in a sample time of 0.5 seconds. This was a relatively good approximation compared to the actual timestamps. In Mission Planner, one could see the exact timestamp for the signals and see that the time between the data points varied between 0.4 to 0.5 seconds. By studying Fig. 7, the undersampling of the pitch motion was an acceptable simplification as the graph for the undersampled data almost matched the original. The smooth transition of pitch change was lost at several time stamps, for example at $t = 0$ s and $t = 5$ s in Fig. 7. The undersampling might have affected the coefficients in the transfer function, but not significantly.

After using system identification in MATLAB with the data from the flight test, two transfer functions had the best fit, and θ_1 was chosen for PID tuning. It has a better fit and a smaller error than transfer function θ_2 , which is why it was chosen. The number of poles was in accordance to what was expected in (5), where the servo response time was taken into account.

The PID that could be acquired for the pitch was able to fulfill the criteria that were set for the controller, with a fast rise time and settling time, and overshoot that fulfilled the desired specifications. On the other hand, the values for the obtained gains were large and negative. The negative sign can be disregarded as this depends on the system's sign convention and the signals that were used. Regarding the values, they are in general smaller, as seen in other cases [4], [7], [21]. However, the values for the gains depend on the system. The system used in this study also had a different order compared to other studies, which have worked with second-order transfer functions. As the transfer function considered the servo's response, the PID for the pitch looked different when tuning. When using smaller gains, oscillations in the response were seen, which is an undesirable behavior in the

system. When tuning, it was noticed that a faster response time was the solution to prevent the oscillation. However, one should also have in mind that the transfer function represented a PWM signal as input compared to other cases where the transfer function is using the deflection angle of the control surface as an input. As mentioned earlier, converting from PWM signals to deflection angles was not possible with the used hardware. This was mainly due to differences that could occur between expected servo angle and actual deflection when working with Clouds.

C. PID for Pitch Rate

The number of data points for the pitch rate was similar to the number of pitch data points, and 20 data points were used to model the pitch rate. The sample time was also 0.5 seconds, which was also a good approximation as the pitch rate was recorded at the same time as the pitch angle. However, the issue with undersampling the pitch rate lead to data that did not represent the pitch rate well as shown in Fig. 8. This showed a case where the system identification method had the drawback of requiring the same number of data points for the input and output and that they needed to be recorded at the same time. In addition, Fig. 8 shows that the pitch rate changed drastically between positive and negative values. Due to the fast-changing values, a sample time of 0.5 seconds was too slow to record a good data set for the pitch rate. In this case, the simplification might not have been suitable to find the transfer function and required a faster sample time.

The transfer function that was chosen as the best for PID tuning was q_1 . Compared to the fit that was acquired for the pitch's transfer function, q_1 only had a fit of 69%. This was a significantly worse result for modeling the pitch rate and was probably caused by the simplifications that were made. Even though the fittings were relatively bad and q_2 with 76% fit could also have been chosen, q_1 was chosen due to its order that matches better with what was expected as seen in (4). The root mean square error of 7.21 degrees/s was quite a large error in speed. This probably showed that the model was not a good model for the pitch rate with the data from the flight test.

Finding the PID for the pitch rate that would fulfill the requirement with no steady-state error was a challenge. As shown in Fig. 12, the response has a steady-state error and stayed under 1 deg/s past the settling time. As shown in Table II, only an integral controller was needed to tune the response for the pitch rate. However, what type of controller is needed depends on the system acquired. The question that should arise is whether the modeled system is accurate or not. The gain was also large and negative, which together with the gains in Table I probably depended on the system rather than on other errors. In general, this controller might not be the best and is probably not suitable for the UAV due to large errors and bad modeling.

D. General Evaluation of the Method

In general, the method used in this report has a good potential for being a suitable method for tuning a PID for a

UAV. This is mainly due to its simplicity and time efficiency when wanting to find an optimized PID.

Regarding the flight tests, the flight logging proved to be helpful in order to know where to find the data. Flight protocols enabled a streamlining of the flight testing procedures. A small team of three proved to be sufficient for flight testing. Nonetheless, the method lacked in clearly establishing which specific flight tests should be conducted, and in what order, as discussed in section VIII-A.

Regarding the PID, even though the method is simple and quick, the simplifications that were needed in order to find a transfer function presented some drawbacks of this method. Unsynchronized and undersampled data sets led to a result that may be unreliable when it came to the coefficients in the transfer functions and the time specifications of the PID. An improved data sampling could have shown the true pitch motion and pitch rate, which were lost when removing data points. On top of that, the speed of pitch changes depending on PWM could have been more accurate and might have shown some time delays in the system response, which were potentially lost with the undersampling. As shown in Fig. 7 and Fig. 8, no delays are apparent. The main reason why the data set needs to be correct and improved is that a bad data set did not give the best and most accurate model for the pitch motion. This thereafter affected the PID gains that were found from the PID tuner.

The transfer functions that had the best fit had the correct number of poles, but not the correct number of zeros if one assumed the servo's transfer function was included. It should be noted that the equations on which the theory is based are simplifications. The real transfer function for the longitudinal equations of motions may contain more poles and more zeros.

In this case, the system identification was relatively suitable. Other reasons why the system identification was not perfect, were that the fit for the pitch and pitch rate was under 90%. A fit better than 95% would allow one to believe the models were accurate with correct coefficients and small errors. As mentioned earlier, a correct model is needed in order to have a stable system, and the PID depends on the transfer function. The system identification method was highly time-efficient and multiple transfer functions could be tested to see how well they fitted the data in a matter of seconds. Compared to analytical methods, using the system identification toolbox was a time-efficient method. An analytical method with simulation could have given a more accurate model, but as discussed in section IV-A, it is not always possible to achieve correct and complete aerodynamic parameters from simulations alone.

The PID tuner was also easy to use and could be obtained within a few minutes. If given a correct and known transfer function, the gains could have been accurate. There is room for questioning whether the obtained values are correct considering they differ from other similar studies. In this study, there are strong suspicions that the values may be incorrect. It was difficult to determine whether they were correct or not since getting a good response was always possible when tuning in the application. Additional flights should have been performed in order to acquire a wide variety of data. The variation would work as a validation that would give a more accurate and

robust transfer function. Another validation process would be to test the robustness of the PID by applying it to the aircraft and analyzing how well the aircraft responds with the PID. The robustness would also be tested as the aircraft's pitch motion can be affected by disturbance and turbulence. These flights can also be used to iteratively fine-tune the gains until a desirable response has been achieved.

E. Future Improvements

To perfect the method and obtain the best possible results for the ALPHA UAV, several improvements are suggested.

First, data collection should be improved. The data sampling depends on the source code that is used by Ardupilot. The source code for Ardupilot is publicly available, which allows changing and synchronizing the sampling rate for the different sensors in the aircraft. A faster sample time would give an accurate model that records the small and drastic changes in pitch and pitch rate. A synchronized sampling between the PWM signals and the pitch angle helps to give a more accurate response time where the output's response might actually be different from what has been presented in Fig. 7 and 8.

Second, while accurate data help to find an accurate model for the pitch motion and pitch rate, data need to be collected more than once. Indeed, collecting data once may give a transfer function that is not robust and only coincidentally matches with the data from the flight test. To ensure the model is accurate and robust, more than one flight test is needed. They would also need to be conducted at different times of day and weather conditions to account for a wide range of aerodynamic disturbances. A robust model would then fit all different cases. This group proposes that at least three different test flights be conducted where pitch changes are recorded, to maximize the accuracy of the transfer function model.

Third, manual or FBWA flight tests should be prioritized before flight tests for autopilot functions. Flight tests should be clearly organized and the aspects to be tested should be determined in advance for more efficient tests.

Fourth, the PID gains should also be validated, which was not done in this study. After finding the PID gains from the PID tuner in MATLAB, the gains can be implemented in Ardupilot. A flight test can then determine how well the controller performs. After conducting flight tests in different conditions, the flight data can be used to analyze the time domain specifications for the controller and its robustness. If the controller does not perform as expected, new PID gains can be found again from MATLAB. However, there are risks involved with testing the PID in this way. If the system is unstable due to the controller, the aircraft is at risk of crashing. Therefore, the PID should be tested in MATLAB or other similar environments before being implemented. Testing for different input signals will show if the PID is robust or not. After these tests, the PID can be implemented into the aircraft. To further prevent crashing, a professional or experienced pilot should fly the aircraft in order to be able to take manual control in case the PID is unstable.

Fifth, an additional comparison could be made between the built-in PIDs in Ardupilot and the PID obtained through

system identification. This would require converting the PWM signals to angles in degrees. If the system identification PID is better than the built-in one, it is pertinent to implement this study for the half-scale and the full-scale ALPHA UAV.

Sixth, flight test data could be filtered in order to reduce noise and uncertainties [4], [5], [11].

IX. CONCLUSION

In conclusion, using system identification and PID tuner with MATLAB toolboxes have the potential to simplify the process of finding a PID for a UAV, thanks to its simplicity and time efficiency. However, in this study, it has been identified that a good set of flight data is needed to find an accurate transfer function as a model. An accurate transfer function will give good PID gains that will allow the controller to perform as expected. The proposed suggestions to improve the quality of the flight data and the model are to increase the sampling rate and perform various flight tests, before and after tuning, in different conditions. The various flight tests will allow the modeling of a robust transfer function and PID controller, and will also validate the model and PID gains. Other additional future improvements would be to compare the PID from this method with the built-in PID in Ardupilot to evaluate whether results from this study can be implemented on the ALPHA UAV.

APPENDIX FLIGHT LOGGING

ACKNOWLEDGMENT

The authors would like to thank Mykola Ivchenko for his support, valuable insights, and patience with our ever-changing focus. We would like to thank Erik Barsby and Casper Augustsson Savinov from group L4a for the help in understanding Ardupilot and for being a great team to work with on the flight tests. We would also like to thank Andreu Matoses Gimenez for giving advice and engaging in useful discussions that helped us in coming up with the method. Finally, we would like to thank group L5; and other members of ALPHA.

REFERENCES

- [1] E. W. Frew, J. Elston, B. Argrow, A. Houston, and E. Rasmussen, "Sampling severe local storms and related phenomena: Using unmanned aircraft systems," *IEEE Robotics Automation Magazine*, vol. 19, no. 1, pp. 85–95, 2012.
- [2] ALPHA. (2022, Jan.) Alpha. [Online]. Available: <https://www.kth.se/alpha>
- [3] M. Ivchenko, *Requirements for the ALPHA UAV v1*. Internal KTH ALPHA document, KTH, Stockholm, Sweden, Nov. 2021.
- [4] M. Schmekel and L. Ringaby, "Simulation and control system design for autonomous gliding to a given location," BSc thesis, KTH, Stockholm, Sweden, May 2021.
- [5] N. V. Hoffer, "System identification of a small low-cost unmanned aerial vehicle using flight data from low-cost sensors," *All Graduate Theses and Dissertations*, 2014. [Online]. Available: <https://digitalcommons.usu.edu/etd/4274>
- [6] L. Ljung, T. Glad, and A. Hansson, *Modeling and identification of dynamic systems*, 2nd ed. Lund: Studentlitteratur AB, 2016.
- [7] R. C. Nelson et al., *Flight stability and automatic control*, 2nd ed. New York: WCB/McGraw Hill, 1998.

- [8] R. Mariani, “Stability,” Lecture notes in Aerodynamics, KTH, Stockholm, Sweden, Feb. 2021.
- [9] HOOAH Aviation Technology CO., LTD. (2016, Dec.) Fpv survey clouds. Changzhou, China. [Online]. Available: <http://www.x-uav.cn/en/content/?474.html>
- [10] C. Augustsson Savinov and E. Barsby, “Building a fixed wing autonomous uav,” BSc thesis, KTH, Stockholm, Sweden, May 2022.
- [11] J. Shen, Y. Su, Q. Liang, and X. Zhu, “Calculation and identification of the aerodynamic parameters for small-scaled fixed-wing uavs,” *Sensors*, vol. 18, no. 1, 2018. [Online]. Available: <https://www.mdpi.com/1424-8220/18/1/206>
- [12] L. Ljung, *System identification theory for the user*, 2nd ed. Upper Saddle River, N.J.: Prentice Hall, 1999.
- [13] The MathWorks Inc. (2022, Apr.) System identification toolbox. [Online]. Available: <https://se.mathworks.com/products/sysid.html>
- [14] The MathWorks Inc. (2022, Apr.) Control system toolbox. [Online]. Available: <https://se.mathworks.com/products/control.html>
- [15] A. J. Keane, A. Söbester, and J. P. Scanlan, *Small unmanned fixed-wing aircraft design: a practical approach*, 1st ed. New York: John Wiley & Sons, 2017.
- [16] A. Sobron, D. Lundström, R. Larsson, P. Krus, and C. Jouannet, “Methods for efficient flight testing and modelling of remotely piloted aircraft within visual line-of-sight,” in *31st Congress of The International Council of the Aeronautical Sciences (ICAS)*, Belo Horizonte, Brazil, Aug. 2018.
- [17] ArduPilot Dev Team. (2022, Feb.) Ardupilot. [Online]. Available: <https://ardupilot.org/ardupilot/index.html>
- [18] ArduPilot Dev Team. (2022, Mar.) Mission Planner Home. [Online]. Available: <https://ardupilot.org/planner/index.html#home>
- [19] ArduPilot Dev Team. (2022, Apr.) FBWA Mode (FLY BY WIRE_A). [Online]. Available: <https://ardupilot.org/plane/docs/fbwa-mode.html>
- [20] The MathWorks Inc. (2022, Apr.) Compare. [Online]. Available: <https://se.mathworks.com/help/ident/ref/compare.html>
- [21] A. E. Onuora, C. C. Mbaocha, P. C. Eze, and V. C. Uchegbu, “Unmanned aerial vehicle pitch optimization for fast response of elevator control system,” *International Journal of Scientific Engineering and Science*, vol. 2, pp. 16–19, 2018. [Online]. Available: <http://ijses.com/wp-content/uploads/2018/03/577-IJSES-V2N3.pdf>

Electric Propulsion for a High Altitude Unmanned Aerial Vehicle

Jacob Friderichsen and David Jönsson

Abstract—The catalogue of observational platforms for space and atmospheric research can be expanded by utilising drones equipped with specialised instrumentation and capable of flying at high altitudes. In this project the requirements for an electric propulsion system applicable to the KTH Royal Institute of Technology ALPHA project are evaluated. A selection of electric motors, propellers and electronic speed controllers are tested to analyse their applicability using two different test setups. The tests include evaluation of propeller characteristics such as thrust and torque generation along with operational angular velocity ranges and the efficiency of brushless DC motors. The results are analysed and extrapolated to approximate the performance in the dynamic environment that the ALPHA aerial vehicle will encounter. From the tested hardware a propeller with a diameter of ten inches and a pitch of seven inches is found to fulfil the requirements. Out of the tested motors, six of them achieve the necessary performance and these are presented with suggestions for further analysis.

Sammanfattning—Mängden observationsplattformar inom rymd och atmosfärisk forskning kan utvidgas genom att använda drönare utrustade med specialiserade instrument som är kapabla att flyga på hög höjd. I detta projekt evalueras kraven på ett elektriskt drivsystem tillämpningsbart på KTH Royal Institute of Technology ALPHA projekt. Ett urval av elektriska motorer, propellrar och elektroniska hastighetsregulatorer testas för att analysera deras tillämplighet genom användning av två olika testuppställningar. Testerna inkluderar evaluering av propellrars dragkraft, vridmoment samt operativ spann av vinkelhastighet, men även börstlösa DC motorers effektivitet och prestanda. Resultaten analyseras och extrapoleras för att approximera prestanda i den dynamiska miljön som ALPHA drönaren kommer att möta. Av den testade hårdvaran uppfyller en propeller med en diameter på tio tum och en stigning på sju tum kraven. Av de testade motorerna uppnår sex av dem den prestanda som krävs och dessa presenteras med förslag på vidare analys.

Index Terms—Electric Propulsion, UAV, ALPHA, Electronic Speed Controller, BLDC Motor, Propeller.

Supervisors: *Nickolay Ivchenko*

TRITA number: *TRITA-EECS-EX-2022:161*

I. INTRODUCTION

The upper atmosphere is an environment that hosts spectacular events such as upper atmospheric lightning described in [1] and auroras, which fall under the category of space weather according to [2]. One category of upper atmospheric lightning has been the subject of studies conducted from the International Space Station in [3] which emphasise that the upper atmosphere and the phenomena therein are of great scientific interest. The ALPHA project at KTH Royal Institute of Technology hopes to expand the existing catalogue of platforms for space observations with an unmanned aerial

vehicle (UAV) capable of flying at high altitudes and of collecting data and images of the sky above, detailed in [4]. This platform will be able to contribute to the research of upper atmospheric phenomena, as well as improve flight following capabilities pertaining to rocket launches.

To be able to photograph upper atmospheric phenomena the ALPHA drone must fly above the clouds. Generally clouds reach around 13 kilometres of altitude in temperate regions as stated in [5]. At these altitudes jet streams can be encountered with extreme wind speeds and when these speeds exceed 60 knots the wind is classed as a jet stream according to [5]. The 60 knots translates to roughly 31 meters per second and to be able to operate in these conditions the ALPHA UAV should be able to fly at altitudes up to 15 kilometres and at speeds up to 40 meters per second. In order to reach these speeds and to climb to this altitude the ALPHA UAV must be equipped with a powerful propulsion system. As stated in [4] the propulsion system is chosen to be electric, with brushless DC motors and Li-Ion batteries driving wing mounted propellers.

High altitude UAV's have seen accelerated development in recent years with projects such as the electric Airbus Zephyr described in [6] and the hydrogen fuelled Boeing Phantom Eye in [7]. Remote controlled drones have also become a widespread hobby and the commercial power electronics for drones have been specialised in this area.

Due to the recent popularity of electric drones, their power electronics have been widely studied and developed. A study of brushless DC motor characterisation specifically for UAV application has been conducted in [8] and another study has modelled and tested electronic speed controllers (ESC) for brushless DC motors in [9].

In the evaluation of power electronics for ALPHA, the aim is to evaluate a selection of commercially sourced electric motors, ESC's and propellers in order to obtain performance data of these products and analyse their applicability to ALPHA.

II. ELECTRONICS THEORY

Electric brushless DC motors work on the principle of electromagnetism and consist of a rotor and a stator. The rotor is, as the name suggests, the rotating part of the motor with a number of permanent magnets of alternating orientation placed along the circumference of the rotor. The stator consists of a number of wound coils capable of producing a magnetic field when subjected to an electric current. The stator coils are activated in three groups by a three phase sinusoidal alternating current and this generates a rotating magnetic field which repels and attracts on the permanent magnets, turning the rotor. The three phases are applied with a phase difference of 120

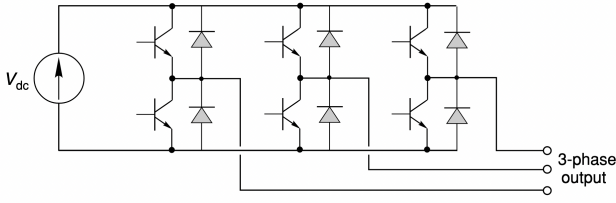


Fig. 1. Schematic of a three phase inverter circuit from [10] chapter 2, pp. 69. This schematic demonstrates the basic principle of an ESC for a brushless DC motor.

degrees so when one group of electromagnets are attracting the permanent magnets of the rotor, another group is neutral and the last group is pushing the rotor. This pattern alternates around the stator due to the sinusoidal nature of the current applied and results in the motor spinning. The rotation speed is proportional to the frequency of the voltage applied since this sets the rotational rate of the magnetic field from the stator. The number of coils on the stator is inversely proportional to the rotation speed of the motor since more coils mean that more periods of the electric signal has to go through the coils in order for the rotor to make a full rotation. The magnitude of the magnetic flux generated is proportional to the voltage applied, as stated in [10]. For a comprehensive overview see [11]. Each commercially available brushless DC motor usually provides a Kv-rating, which is a measure of motor RPM (Rotations per minute) per volt applied. For example a motor without load and a Kv-rating of ten rotates at a rate of ten RPM when one volt is applied as described in [12].

The propulsion system is provided with a DC current so in order to have the motor run with a three phase sinusoidal current, the signal has to be processed. The component responsible for this is the ESC, which is a key component in this type of propulsion system. The basic working of an ESC is as a three phase inverter circuit and a simple schematic is shown in Fig. 1. The transistors in this circuit are of the type metal-oxide-semiconductor field-effect transistor (MOSFET) and are activated in such a way that short pulses of correct amplitude and polarity are generated. The pulses are the basis of three pulse width modulation (PWM) signals that respectively sum up to the required sinusoidal current required as described in [10]. A micro controller in the ESC handles the activation of the MOSFET's so the desired three phase voltage is achieved by either measuring the back electromotive force (EMF) or receiving Hall-effect sensor information, from the motor. This provides positional information of the rotor and with an algorithm the micro controller determines the activation sequence of the MOSFET's, described in [11]. To be able to throttle the motor a control signal, also of the PWM type, is sent to the ESC micro controller. A PWM signal of width $1000 \mu s$ corresponds to 0% throttle and a signal of width $2000 \mu s$ corresponds to 100% throttle. With this input the micro controller regulates the frequency of the output three phase current so that the motor speed corresponds to the desired throttle setting as explained in [13].

ESCs often offer the option to adjust the timing of the three phase signal sent to the motor. This timing refers to a phase shift of all three phases of sinusoidal electric signals which

leads to an advance in the rotating magnetic field in the motor. This is due to the coils being excited at a specified angle before the corresponding permanent magnet of the rotor is above the respective coil. For example a timing of 12 degrees corresponds to each coil in the active group being energised when the permanent magnet of the rotor is 12 degrees away from being over the related coil. A high timing setting can increase motor RPM for a given throttle setting but also results in decreased efficiency and higher motor temperatures. The risk of a high timing setting can also be that the rotor is unable to synchronise with the rotating magnetic field and the motor stops, all described in [14] and [15].

III. AERODYNAMICS THEORY

The propulsion system relies on propellers for generating thrust and even though this is not the main focus of this project, some aerodynamic concepts are crucial to interpreting requirements and results. A normal aerofoil generates lift according to the equation

$$L = \frac{1}{2} \rho_{\infty} V_{\infty}^2 S C_L \quad (1)$$

from [16]. Here L is the lift force generated when a fluid is flowing over the aerofoil, ρ is the density of the fluid and V is the fluid velocity over the aerofoil. The infinity subscript refers to the fact that these values are the free stream characteristics of the fluid flow. The S is the area of the aerofoil and C_L is the coefficient of lift, that depends on the aerofoil features, such as angle of attack (α). Propellers consist of a number of rotating aerofoils and the force the propeller generates in total is denoted thrust. A propeller generates a certain amount of thrust for a specific RPM, regardless of which motor is used to drive it. The thrust generated depends on the propellers geometry, RPM, forward air speed as well as air density. The aerodynamics of propellers can be complicated and the subject of blade element theory is beyond the scope of this project. Instead an equation for the static thrust from a propeller was described in [17] with the equation

$$T = k_T \rho n^2 D^4 \quad (2)$$

where T is the thrust force generated by the propeller, ρ is the air density, n is the angular velocity, D the propeller diameter and k_T a thrust coefficient. Since practical tests in this project only consider static thrust, the ambient air speed does not need to be considered until later. Note that the thrust is dependent on the square of the angular velocity.

The notation in this project describes propellers in terms of their diameter and pitch. Propellers are described as 'diameter x pitch' with the measurements given in inches. As an example, one propeller used had a diameter of eight inches and a pitch of 4.5 inches and are henceforth described as the 8x4.5 propeller.

The relation between mechanical power (P), angular velocity (ω) and torque (L) is

$$P = \omega L \quad (3)$$

where ω is the angular velocity in *radians/s*. The formula can be rewritten to use RPM instead:

$$P = 2\pi/60 \cdot RPM \cdot L \quad (4)$$

These equations are used to calculate the mechanical power and is in turn used to calculate the efficiency of both motor and propeller. According to [17] the thrust and torque of a propeller can be modelled as described in equation (2) and

$$L = k_L \rho n^2 D^5 \quad (5)$$

where n is a measure of propeller rotation speed. The coefficients k_T and k_L are defined by the propellers design. Combining these equations shows that any given propeller has a linear relation between the thrust and torque such that:

$$k_p = L/T = \frac{k_L}{k_T} D \quad (6)$$

$$L = k_p T \quad (7)$$

which is in theory independent of rotational speed, air density and other factors than diameter. The value k_p is a propeller specific constant. This linear model does not account for non-lift related drag on the propeller and in reality a propeller travelling at high forward speed with its pitch speed equal to its airspeed generates zero thrust, but a non-zero torque, breaking this relation. This torque stems from drag on the propeller due to its rotation through the air and is not negligible, but low in relation to the drag induced by lift when referring to [18]. The linear model is deemed as an acceptable compromise, but more work should ideally be done to complete the model.

The concept of dynamic thrust is the reduced thrust effect from a propeller spinning when moving in a relative airflow. The tests conducted in this project studies the static thrust of the propulsion system, but when the system is mounted on the UAV the forward speed changes the thrust performance of the propeller. When the propeller is not moving forwards the relative airflow over the propeller blades is directly opposite to the blades direction of motion. When the propeller moves forward through the air the relative airflow over the blade has a component opposite the propellers direction of travel and thereby decreases the propeller blades effective angle of attack, as described in [19]. An illustration of this is provided in Fig. 2 where the relative airflow is shown with three arrow heads, with components opposite the direction of the horizontal and vertical arrows.

In order to analyse the propulsion performance at different forward speeds and altitudes a model of dynamic thrust needs to be considered. Referring to [20] and [21] thrust can be approximated as decreasing linearly to zero as the forward velocity of the aircraft approaches the pitch velocity

$$V_{pitch} = pitch[in] \cdot RPM \cdot 0.0254 \frac{m}{in} \frac{1}{60} \frac{min}{s} \quad (8)$$

of the propellers. Propellers are aerofoils and as such they follow the lift equation (1) which means that the thrust of the propeller is proportional to the density of the atmosphere. The following formula for dynamic thrust at altitude

$$T_D = \frac{\rho}{\rho_{SL}} \left(1 - \frac{V}{V_{pitch}}\right) T_S \quad (9)$$

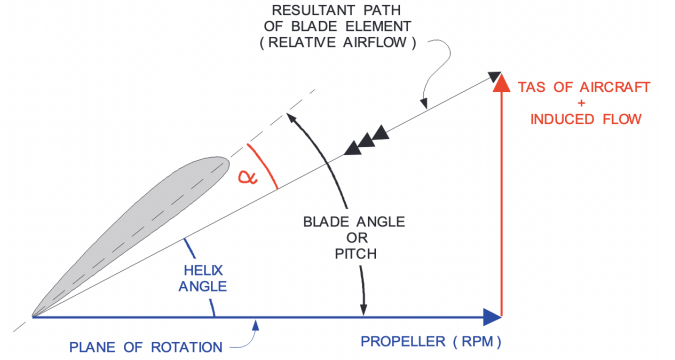


Fig. 2. Relative airflow over a propeller blade. TAS refers to true air speed and denotes the forward velocity of the propeller and α denotes the effective angle of attack of the propeller blade. From [19] chapter 15 pp. 491.

is proposed, where T_S is the static thrust at sea level and ρ_{SL} the air density at sea level. Using equation (2) static thrust is modelled as proportional to the second power of the angular velocity. Simplifying the static thrust formula in equation 2 to

$$T_S[gf] = C_T \cdot RPM^2 \quad (10)$$

which when solving for RPM gives

$$RPM = \sqrt{\frac{T_S[gf]}{C_T}} \quad (11)$$

where $C_T[\frac{gf}{RPM^2}]$ is a propeller specific coefficient describing its static performance at sea level. This coefficient can be experimentally determined for any propeller. Inserting equation (8) and equation (10) into equation (9) gives

$$T_D[gf] = C_T \cdot RPM^2 \frac{\rho}{\rho_{SL}} \cdot \left(1 - \frac{V \cdot 60 \text{ s} \cdot \text{in}}{pitch[in] \cdot RPM \cdot 0.0254 \text{ m} \cdot \text{min}}\right) \quad (12)$$

Using the quadratic equation, equation (12) can be solved for RPM.

$$RPM = F(V, pitch[in]) + \sqrt{F(V, pitch[in])^2 + \frac{T_D[gf] \rho_{SL}}{C_T \rho}} \quad (13)$$

where

$$F(V, pitch) = \frac{V \cdot 30 \text{ s} \cdot \text{in}}{pitch[in] \cdot 0.0254 \text{ m} \cdot \text{min}} \quad (14)$$

In order to fly at a stable speed the total drag of the aircraft should be equal to the total thrust. Drag force is calculated with the drag equation from [16]

$$D = \frac{1}{2} \rho V_\infty^2 C_d S \quad (15)$$

with elements as described earlier. Note that the area S refers to the total frontal projection of the aircraft.

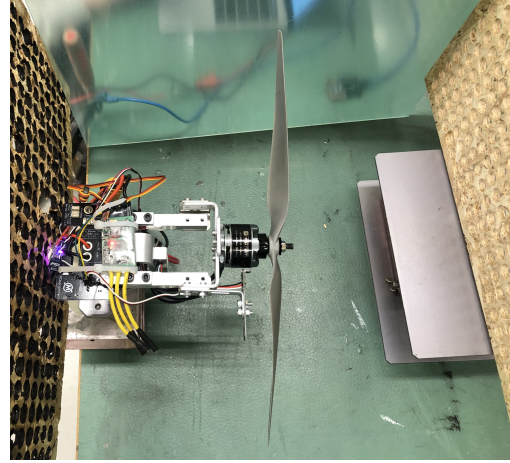


Fig. 3. Experimental setups for testing of electronics. The first setup with the Turnigy thrust stand is shown on the left with associated equipment. The second setup with the Tyto Robotics series 1585 thrust stand is shown on the right.

IV. METHOD

To find the optimal power electronics for the ALPHA UAV, motor testing was conducted on multiple electric motors, ESCs and propellers recovered from past projects in order to get experience and data to base the power electronics requirements on. Testing was done with two different test setups. The first setup used the Turnigy thrust stand and power analyser V3. Using this setup, all data was manually read and inserted into a spreadsheet. The values read were thrust, current, voltage and RPM. Eleven linearly distributed data points were generated from 0-100% throttle. The second setup used the Tyto Robotics series 1585 along with the accompanying RCbenchmark software. This setup allowed automated reading of thrust, torque, rpm, current and voltage. In these tests up to 70 data points were generated for each test in a fraction of the time it took using the first setup. Both setups used 12 V server rack power supplies with a current limit of 69 A. For 24 V tests, two power supplies were connected in series.

In the Turnigy setup an oscilloscope was used to read and assist in manually tuning the PWM signal being sent from the Turnigy thrust stands throttle. A UN203R digital clamp meter was used to perform current measurements on the positive DC connection from the power supply. A multi-meter was used to measure the voltage over the power supply. Thrust was read directly from the Turnigy test stand. RPM measurements were done with a Peaktech P2790 optical tachometer. All values were then inserted into a spreadsheet template.

For the Tyto robotics setup the measurement process was a lot smoother. The thrust stand was connected by USB to a computer with the RCbenchmark software. An optical RPM sensor attachment was purchased along with the stand and mounted to it. The RCbenchmark software automatically generated a CSV file containing measurements of thrust, torque, RPM, current, voltage, vibration, 3-axis acceleration data and derived power and efficiency data for a range of PWM signals. The tests consisted of a 30 second linear sweep from 0 to 100% throttle.

The laboratory setup of the testing equipment is shown in Fig. 3 with the first setup shown on the left with the Turnigy

thrust stand and power analyser V3 and the second setup with the Tyto Robotics RC benchmark Series 1585 thrust stand on the right.

The Tyto robotics 1585 series thrust stand has built-in safety cutoffs for thrust, torque, RPM, current, voltage, vibration, and electric power. These limits were all set to the lowest specification of the components in each test, ensuring propellers did not over speed and motors did not get burnt or broken by too large forces. This meant that not all motor and propeller combinations could be tested to the motors max throttle, leaving some gaps in the data. Most important for safety was to keep the propellers below their maximum recommended speed as to avoid them shattering. The RPM limits for APC propellers can be found in [22]. APC propellers 8x4.5, 10x4.5, 12x4.5, 14x5.5 and 16x5.5 were of the multirotor (MR) type except for 5x5, 7x5 and 10x7 which were the thin electric (E) type. MR and E propellers are rated at 105000 RPM · in and 150000 RPM · in respectively. This value was then divided by propeller diameter in inches to obtain a propellers maximum RPM. For each test this value was input as a safety cutoff to ensure safety.

A. Propeller Evaluation Method

The propulsion system of the ALPHA UAV consists of propellers driven by electric motors. In order to evaluate the motors and their associated electronics, their performance was tested with different propeller types to get a realistic and general picture of the characteristics. The propellers selected for practical testing were APC Propellers made from a long glass fiber composite as written in [23] and GEMFAN propellers made from poly carbonate as written in [24] and [25]. The propellers from APC Propellers had two blades and were of the types: 5x5, 8x4.5, 10x4.5, 10x7, 12x4.5, 14x5.5 and 16x5.5. The propellers from the GEMFAN brand were of the types: 7x4 with three blades and 5x4 with six blades. These propellers were chosen as they represent a broad range of propeller diameters and offered the opportunity to compare the effects of varied propeller diameter and pitch. To characterise the propeller types used and to compare the two types of test

setups, the propellers were tested to their maximum rated RPM or until the motor could not accelerate any more.

For the tests completed with the Turnigy thrust stand setup, all data points of thrust and RPM were collected for each propeller and second degree polynomial fits were generated with the least squares method to fit these data points. The linear and constant term were set to zero as the thrust is only dependent on the square of the angular velocity and has no linear or constant term. A second degree polynomial fit was chosen due to the thrust being dependent on the square of the angular velocity, that is the RPM, as described in Section III. For the second test setup with the RC Benchmark thrust stand, tests were conducted where each propeller was tested through an RPM range one time.

B. Motor Test Method

To find each available motor's specific performance with a propeller, a series of motor tests were conducted with each available propeller. This provided a measure of torque, RPM, power and motor efficiency for each motor making them easy to compare. The motors tested were the following:

- Leopard Hobby 1600 Kv
- Aerodrive 2836 1500 Kv
- ZOHD MKII 1300 Kv
- Aerodrive 3536 1400 Kv
- Aerodrive 3548 1050 Kv
- Emax Grand Turbo 985 Kv
- Aerodrive 4240 740 Kv
- Aerodrive 4250 500 Kv
- T-Motor 400 Kv

Each motor was tested for all before mentioned propellers that would fit. They were tested on both the first and second setup with both 12 and 24 volts. These motors represent a selection of different brands in the industry with a wide range of Kv-ratings. The results of these tests were the basis for determining if any of these motors could perform as required for ALPHA to conduct its mission.

C. ESC Evaluation Method

In order to optimise the electronics, the ESC should give optimal performance in conjunction with the selected motor and propeller. Among the ESCs available for testing, three were selected to be of significant interest. The ESCs selected were T-motor F45A, Hobbywing Flyfun 30A and YEP 60A. The ESCs that were not selected had significantly lower current limitations and were therefore deemed not applicable. The evaluation of the ESCs comprised of pairing them with a motor and a propeller, and testing the performance of the setup. For this pairing one motor of each major brand available, Turnigy Hobbywing and T-motor, were selected. To analyse the performance characteristics for different propeller types, propellers of dimensions 10x7 and 16x5.5 were picked. The two ESCs were then tested with each motor and each propeller. The timing of the three phase signal was adjusted as a parameter in the evaluation of the ESCs' performance, and as a way to optimise the ESCs' performance. The programmable timing options for each ESC tested are as follows:

- T-motor F45A: An integer range from 0° to 30° and Auto

- Hobbywing Flyfun: 0°, 5°, 8°, 12°, 15°, 20°, 25°, 30°
- YEP: 0°, 6°, 12°, 18°, 24°, 30°, Auto

According to [11] the 'Auto' timing feature makes the micro controller in the ESC select an appropriate timing according to the feedback received from the motor, either from the back EMF or response from Hall-sensors in the motor. All tests were conducted with a power supply of 24 volts.

V. RESULTS AND DISCUSSION

A. Propeller Evaluation

The result of the propeller evaluation tests are presented in Fig. 4 with the produced thrust plotted against the RPM on the left side and torque plotted against produced thrust on the right side. The left plot contains both polynomial fits, generated from the motor tests with the first setup, measurements from the second setup and manufacturer data. The polynomial fits are the dashed lines described in the legend on the right side of the plot and the exact measurements are the filled lines described in the legend on the left side. Data from the propeller manufacturer in [18] is also plotted to compare the results.

The results show that the polynomial fit for the data points generated with the first setup does not exactly coincide with the tests conducted on the second setup and this can primarily be attributed to a difference in calibration. For smaller diameter propellers, the Tyto robotics setup measures a lower thrust for a given RPM than the Turnigy setup, but for larger propeller diameters the graphs become more similar. The manufacturers data is slightly above both polynomial fits and results produced with the Tyto robotics setup. All polynomial fits of thrust vs RPM resulted in an R^2 value above 0,99 as seen in Table I, which confirms that the fit is representative of the data. Variation in data points can be explained by the error sources later discussed. Table I also provides the coefficient for the quadratic term obtained from the regression. The larger diameter of the three bladed propeller means it produces more thrust than the six bladed propeller with a smaller diameter. Comparing the two propellers with a ten inch diameter shows that an increased pitch gives a slight improvement of thrust for a certain RPM. The plot does show a correlation between the measurements made on the two different setups and the data from APC propellers [18]. Therefore both setups were deemed fit to provide reliable data for evaluation.

The right plot in Fig. 4 shows the torque characteristics of all the tested propellers, measured on the second setup. Torque is plotted against thrust and the expected linear relation is seen. Linear regressions of the data points for each propeller are also shown in the plot along with the data from APC propellers [18]. From the right plot it can be seen that the biggest propeller 16x5.5 generates the highest torque for a given thrust, which is to be expected since this propeller has the largest diameter according to equation 6. An interesting result is that the three bladed propeller generates the lowest torque for a given thrust, suggesting it is an efficient design. By closely inspecting the graph it can be determined that the six bladed propeller with a smaller diameter requires higher torque than the three bladed propeller to generate the same amount of thrust. This displays that propeller characteristic

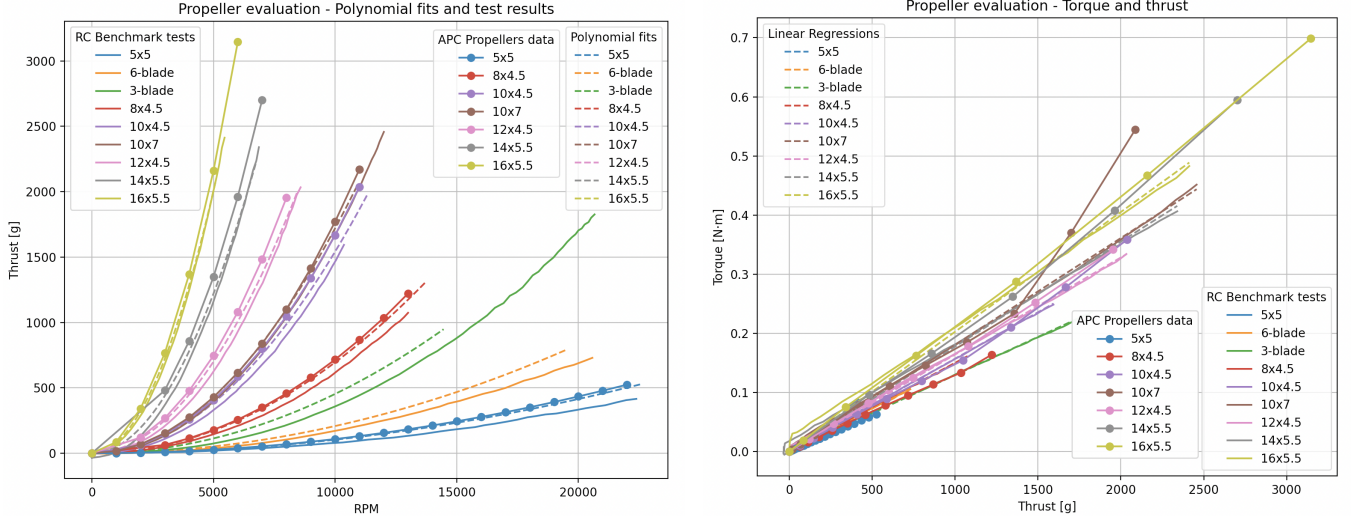


Fig. 4. Propeller evaluation plots. On the left are graphs from the second setup and polynomial fits of the test data from the first setup, showing the relation between RPM and thrust for each tested propeller. On the right the relation between thrust and torque is shown for each tested propeller with graphs and linear fits for data collected on the second setup. Both plots also contain graphs of the data collected from the manufacturer of the APC propellers.

TABLE I
PROPELLER POLYNOMIAL FIT PARAMETERS C_T (EQ.10) FOR EACH PROPELLER

Propeller	Coefficient for ω^2	R^2 value
5x5	$1.0360 \cdot 10^{-6}$	0.9966
six blade	$2.0794 \cdot 10^{-6}$	0.9986
three blade	$4.5349 \cdot 10^{-6}$	0.9960
8x4.5	$6.9501 \cdot 10^{-6}$	0.9977
10x4.5	$15.418 \cdot 10^{-6}$	0.9974
10x7	$17.199 \cdot 10^{-6}$	0.9988
12x4.5	$28.006 \cdot 10^{-6}$	0.9964
14x5.5	$48.810 \cdot 10^{-6}$	0.9967
16x5.5	$80.323 \cdot 10^{-6}$	0.9985

are not a simple matter since it in theory should be more efficient due to its smaller diameter. The outlying results of the Gemfan multiblade propellers show that blade number could be interesting to investigate in the future.

All torque vs thrust plots generated with test data from this project appear to be linear. Linear regression was applied to the data for each propeller to determine the slope. The resulting values are presented in Table II along with the attained R^2 value which is appropriate for deeming the linear regression a valid model. The data from the manufacturer however differs in some cases such as for the 5x5 propeller where the slope is significantly lower than the result from this project. This might be due to the motor blocking a significant part of the 5x5 propellers air flow in the tests performed within this project. The most interesting difference is for the 10x7 propeller, which suddenly deviates at around 1400 gf thrust in the APC test. This is a massive deviation from the test results for the 10x7 propeller in this project. The deviation could be a measurement error on APC propellers part, or potentially a sign of flow separation occurring due to the high pitch. Flow separation

TABLE II
PROPELLER LINEAR REGRESSION PARAMETER

Propeller	Coefficient for L	R^2 value
5x5	$2.0398 \cdot 10^{-4}$	0.9951
six blade	$1.4609 \cdot 10^{-4}$	0.9978
three blade	$1.2924 \cdot 10^{-4}$	0.9992
8x4.5	$1.5007 \cdot 10^{-4}$	0.9963
10x4.5	$1.5703 \cdot 10^{-4}$	0.9993
10x7	$1.8057 \cdot 10^{-4}$	0.9992
12x4.5	$1.6412 \cdot 10^{-4}$	0.9982
14x5.5	$1.7753 \cdot 10^{-4}$	0.9954
16x5.5	$2.0259 \cdot 10^{-4}$	0.9951

seems improbable however since the result is not consistent with the measurements in this project.

For the data collected in this project propeller torque is, to a good approximation, linear with propeller thrust and that means that torque is also proportional to the square of the angular velocity. With Table II the impact of propeller pitch can be determined by studying the propellers with a diameter of ten inches. The propeller 10x7 with a pitch of seven inches has a steeper gradient than the 10x4.5 propeller with a pitch of 4.5 inches, which means the propeller with a larger pitch requires a larger torque to generate the same thrust as a propeller with lower pitch. With the two plots in Fig. 4 a static thrust requirement can be translated to a requirement of RPM and torque for a specific motor which aids in the selection of an optimal propeller.

B. Motor Evaluation

From the motor tests, the resulting plots in Fig. 5 are presented with resulting torque on the vertical axis and RPM on the horizontal axis. Each plot has a title describing the motor which is tested and to which the data is applicable. Each

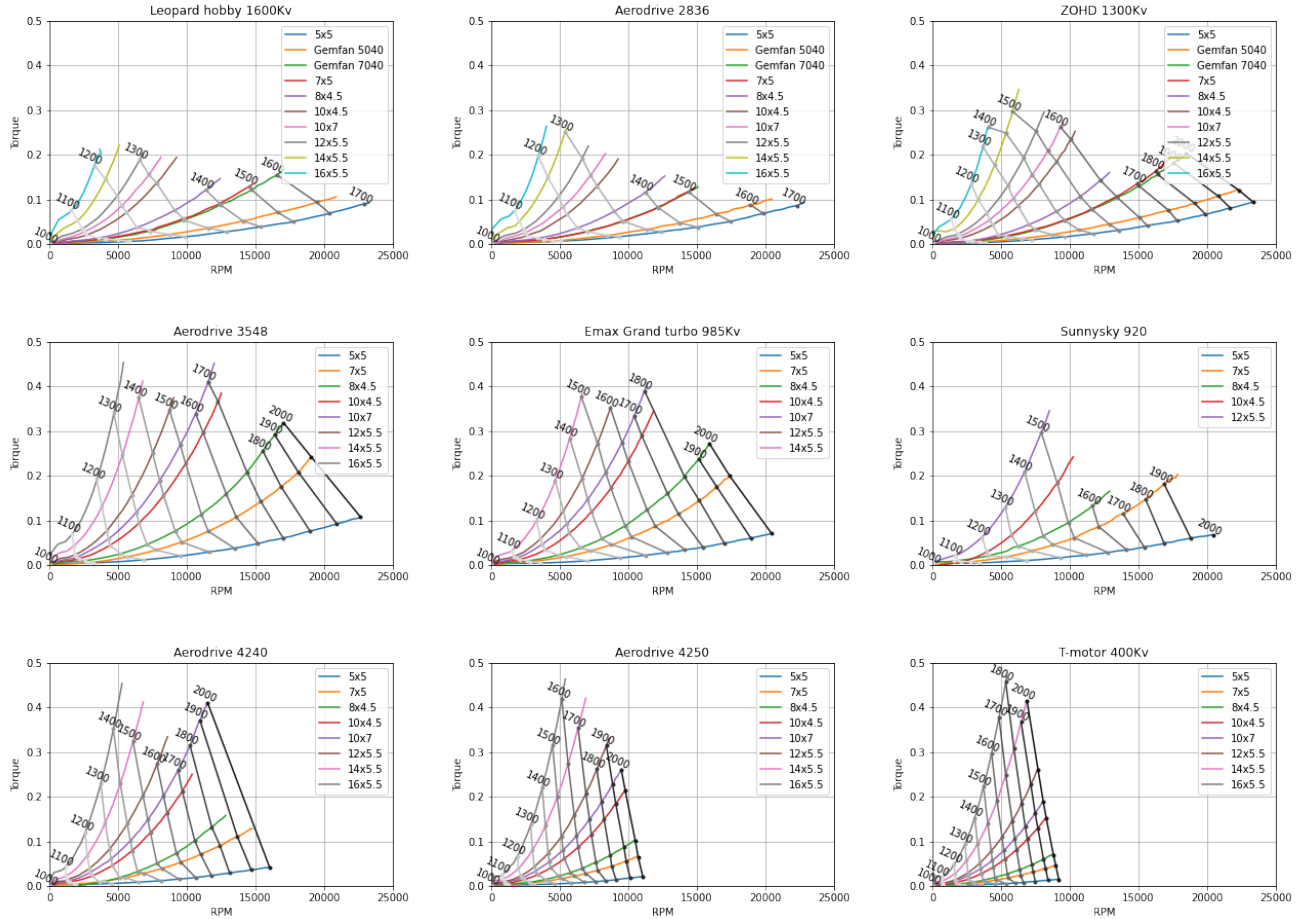


Fig. 5. 24V Torque vs RPM plots with PWM level curves. Coloured lines show torque resistance for each propeller as a function of its RPM. The grey-scale curves show the PWM control signal required to reach a certain torque and RPM.

coloured curve in the plots represents the performance with a tested propeller described in the legend. Additionally the PWM signal setting has been marked to represent the throttle setting throughout the tests. As mentioned the PWM signal goes from $1000 \mu s$, which represents 0% throttle, to $2000 \mu s$ which represents 100% throttle.

It is clear from the plots in Fig. 5 that the tested motors with lower Kv-ratings are generally capable of delivering more torque than the higher Kv-rated motors. This is natural since for a given power, torque is inversely proportional to RPM as seen in equation (4). A slower, lower Kv-rated motor is as a rule of thumb bigger and has a higher power rating than a high Kv motor, which enables higher torque even at the same RPM. The trade-off with low Kv motors is that they cannot spin as fast and hence do not utilise the full performance of small propellers. On the other hand high Kv motors such as the Leopard hobby 1600 Kv motor exceeds their power rating before reaching full throttle, even for the smallest propeller tested which is a bad characteristic. Every motor tested reaches some safety limit that aborts the test before full throttle is reached for at least one propeller. It is important to note that at higher speeds and altitude, the torque exerted on the motor by the propeller is lower due to the lower air density and relative speed between the propeller and the air. A configuration that

cannot reach full throttle on the ground due to power or thrust limitations might be able to do so when flying.

More importantly the plots in Fig. 5 can be used to match a propeller with a motor. With the knowledge of the propellers Torque/thrust ratio and thrust/RPM curve, a torque can be found for a certain RPM of the propeller. If for example 500 gf is required there are many propeller options that fulfil this requirement, but the motor has to be matched with the right propeller. For example T-motor 400 Kv would not be capable of reaching 500 gf with the 8x4.5 propeller since the motor is unable to reach the required RPM. The 12x4.5 propeller according to Fig. 4 requires only approximately 4500 RPM for 500 gf and only 0.09 Nm of torque for 500 gf. Static sea level torque and RPM can also be obtained in an arguably more accurate way using equation (11) and equation (7) in conjunction with Tables I and II. Using this method, 4230 RPM and 0.082 Nm of torque is calculated for the 12x4.5 propeller. With these values of torque and RPM it can be seen in Fig. 5 that T-motor 400 Kv would have need a PWM signal around $1400 \mu s$. This means that the motor is at 40% throttle, which indicates that it is well within its capability. In this case the required PWM value can be read straight from Fig. 5 since the 12x4.5 propellers static, ground level torque to RPM curve is plotted in the figure.

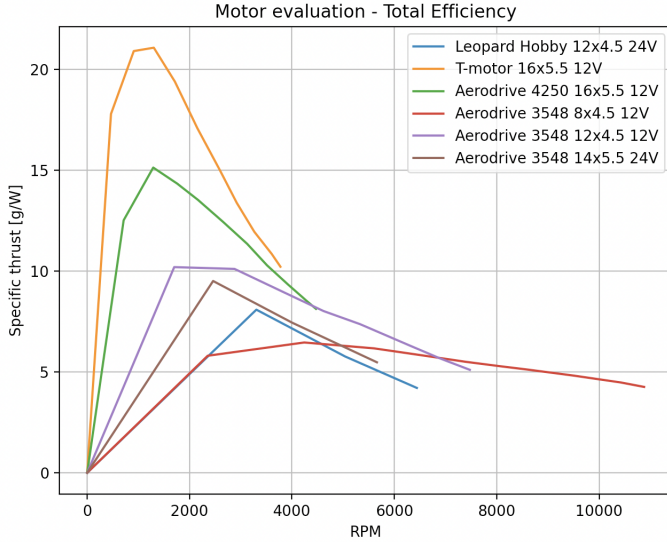


Fig. 6. Traditional efficiency curve with specific thrust plotted against RPM for a selection of tests.

In another example however 500 gf might be needed at a forward velocity of 15 m/s and at an air density $\rho = \rho_{SL}/2$. By referring to equation (13), the required RPM is calculated to be 11100 RPM. The torque required is still 0.082 Nm because of the constant T/L relation in equation (6). When referring back to Fig. 5 it can be seen that this is outside of the T-motor 400 Kv RPM capability. Another motor or propeller would have to be considered in this example.

To evaluate the efficiency of the electric motors, several parameters should be considered. The propeller efficiency is determined by dividing the generated thrust by the mechanical power described by the product of the RPM and torque as

$$eff_{prop} = T/P_{mechanical} \quad (16)$$

where the mechanical power is described by equation (4). The total efficiency of the system also called specific thrust is attained by dividing the generated thrust with the electric power applied. Lastly the motor efficiency is determined by the fraction

$$\eta = \frac{P_{mechanical}}{P_{electric}} \quad (17)$$

given as a percentage. Traditionally the efficiency for a motor and propeller setup is presented as the specific thrust plotted against RPM as shown in Fig. 6. In this plot a small selection of data from the motor tests is shown to demonstrate a traditional way of presenting the efficiency.

In order to separate the motors performance from that of the propeller, motor efficiency is focused on instead according to equation (17). The results of these efficiency evaluations are presented in Fig. 7 with a plot for each tested motor. In these plots efficiency level curves were generated from the same test data as in Fig. 5. Each plot in the figure has a title describing which one of the tested motors the plot displays. The plots can be used to get an approximate efficiency for a motor based on an application with a where the torque and RPM is known. If applied to the previous static example of the T-motor 400 Kv and 12x4.5 propeller at 4225 RPM and 0.082 Nm, it is found

that the motor efficiency is between 70% and 72%, which is decent. The trade-off between torque and RPM capability with different Kv-ratings is very important to consider when choosing motors and propellers. As shown in Fig. 5 the higher Kv motors have a much greater RPM capability, with the drawback being a reduced maximum torque.

C. ESC Evaluation

The ESC tests, conducted with the selected motors and propellers, yielded the plots seen in Fig. 8 where the results of each test are compared and the timings giving the best performance for each case are plotted against each other. The plots represent a selection of the best results from all the tests. The selected results for the Aerodrive 4240 and T-motor 400Kv combined with propellers 10x7 and 16x5.5 are shown in four plots in Fig. 8. To analyse the performance, the thrust generated is plotted against the electrical power consumed. It is apparent from Fig. 8 that the timing does not make a significant difference in thrust performance. The selected best performing timing settings does not differ more than around 50 gf for the Aerodrive 4240 motor with the 10x7 propeller and the 16x5.5 propeller. It should be noted that the variation in all the results for the ESC tests might also partially be a product of error sources in the test setup, like varying air density and temperature. It should also be noted that for the Aerodrive 4240 tests, the T-motor F45A ESC's fixed timing settings marginally outperforms the auto timing setting. However this difference in performance is also small enough, that it could potentially be attributed to sources of errors that are discussed in a later section.

From Fig. 8 it can be seen that the auto timing feature of the YEP 60A ESC performs slightly better than all other selected best performing timing settings, for all three tested ESCs, for the T-motor brand motor. It can also be seen that the two best performing timing settings with the T-motor brand motor and the 16x5.5 propeller, are the auto timing features of the tested ESC's. Tests with the Hobbywing Flyfun ESC are not performed for the T-motor brand motor and the 16x5.5 propeller which explains why the bottom right plot in Fig. 8 lacks the graphs for these tests. For both of these plots it can be seen that the difference in performance is still around 50 gf which is not deemed a significant difference when error sources are accounted for. In general the results show no major performance advantage for any of the ESCs. Based on that, other factors should be considered when choosing an ESC such as its weight, voltage and current limit.

D. Notes on battery requirements

For the testing in this project, power was supplied by server power supplies at a voltage around 12 and 24 volts. From the initial thrust test using the first setup, it was concluded that the performance was significantly better using 24 volts than 12 volts in line with the theory described in Section II. Because of this most of the results presented are based on the data obtained during the 24 volt tests. It should be noted however that from Fig. 6 that the total efficiency of the system is dependent on the voltage supplied. Further more each motor

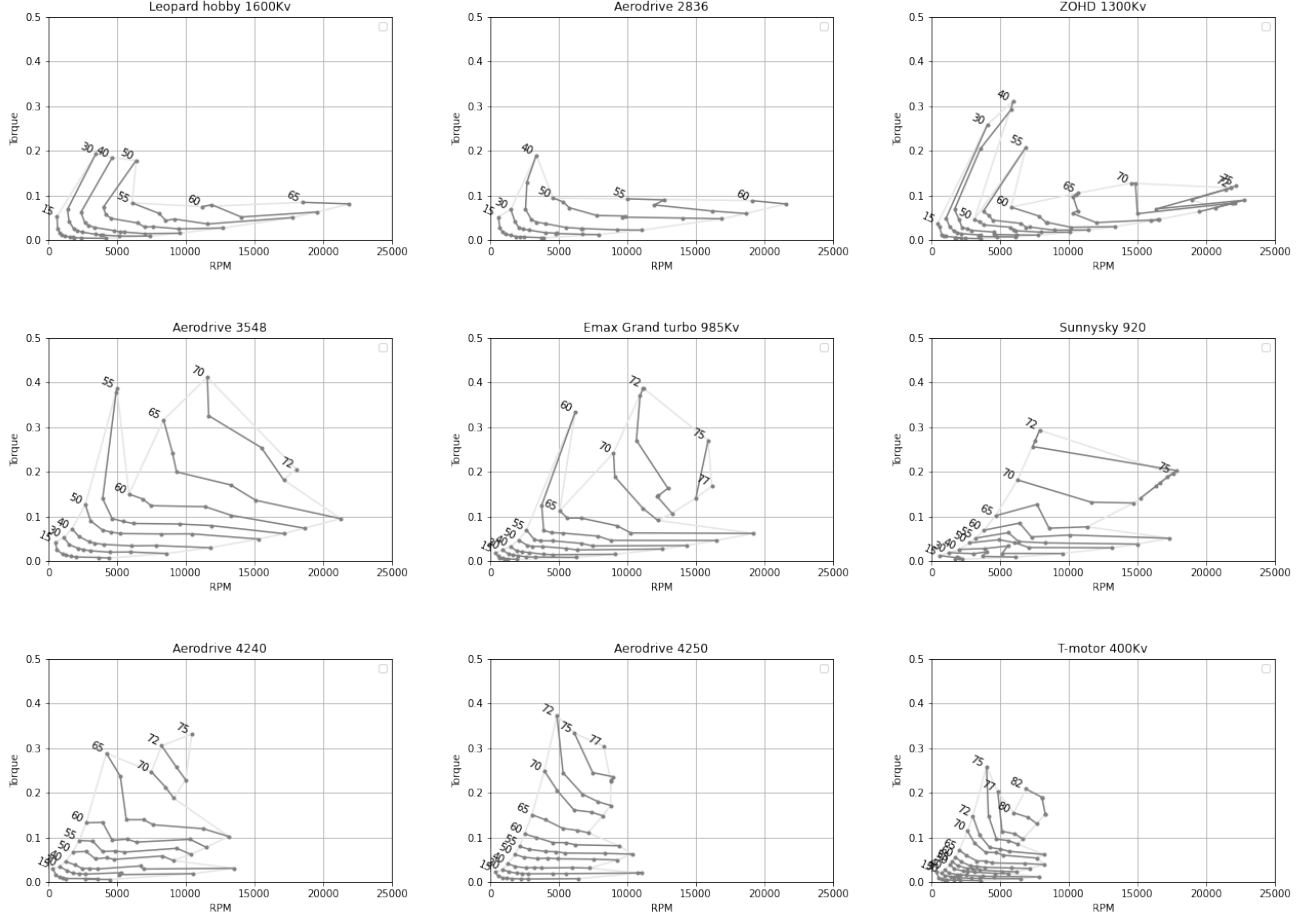


Fig. 7. Plots of efficiency level curves in percent. Curves are plotted in torque vs RPM plots. Data is taken from the 24V tyto-robotics motor test series. Light gray borders have been plotted around the results to mark the areas where efficiency can reliably be assumed to lie between the efficiency curves.

has a maximum operating voltage which should be considered when constructing the propulsion system.

The propulsion system used on the ALPHA UAV will rely on batteries for power. Further tests using commercial batteries with a wider range of voltages would better reveal the performance dependency on voltage. From the theory and tests the only result in regards to voltage for the propulsion system is that the motor RPM and power increases with increased voltage. Something to consider when choosing battery is that the supply voltage decreases as a battery is discharged, which lowers the maximum RPM and power of the motors.

E. Propulsion option for ALPHA

In order to apply the results from this project to the ALPHA UAV a special case is considered. The most extreme plausible use case of ALPHA is flying at 15 km altitude and fighting 40 m/s winds. The purpose of flying at 15 km is to be above the highest appearing clouds, the cirrus clouds. Depending on latitude these clouds can reach a maximum of 13 km in temperate regions, but as much as 18 km in tropical regions as read in [5]. A goal of 15 km is set for the KTH ALPHA project and this should allow it to avoid all clouds in the temperate regions, and with some luck even in tropical regions. Computational fluid dynamics (CFD) analysis by the ALPHA

TABLE III
RPM AND THRUST REQUIREMENTS FOR FLIGHT AT 15 KM AND 40 M/S

Propeller	RPM	Torque [Nm]
5x5	40 150	0.0290
Gemfan 5040	35 570	0.0207
Gemfan 7040	30 100	0.0184
8x4.5	25 906	0.0213
10x4.5	23 440	0.0223
10x7	16 600	0.0256
12x4.5	22 410	0.0233
14x5.5	18 180	0.0252
16x5.5	17 800	0.0288

lightweight structures team generated tables of C_D and C_L for α angles at sea level and altitude seen in the appendix. The weight of Alpha is seven kg as stated in [4], which requires seven kg of lift to sustain altitude. Using equation (1), solving for C_L yields

$$C_L = \frac{2L}{\rho_{\infty} V_{\infty}^2 S} \quad (18)$$

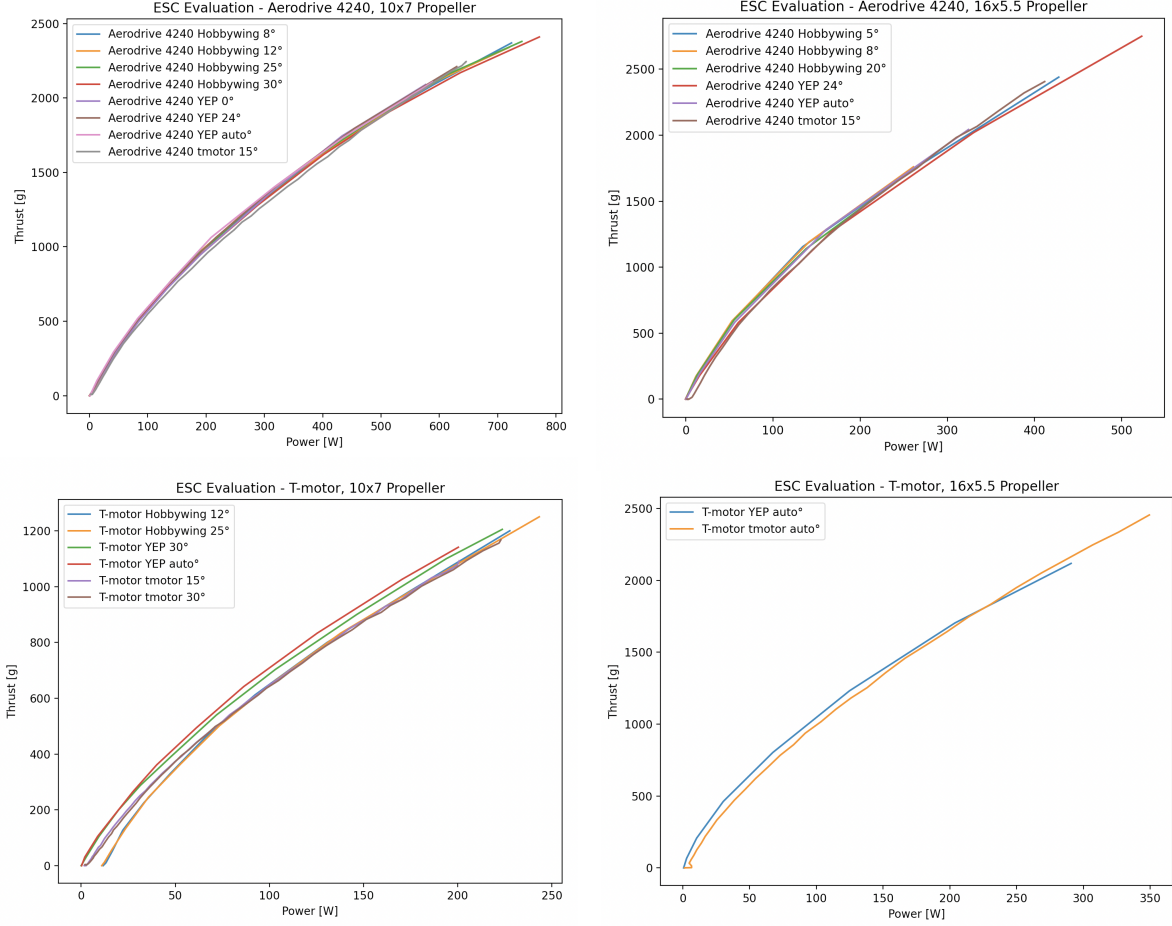


Fig. 8. Results of the ESC evaluation with the test of the Aerodrive 4240 with the 10x7 propeller in the top left corner. The tests of the Aerodrive 4240 with the 16x5.5 propeller are shown in the top right corner. Results of the test of the T-motor with the 10x7 propeller are shown in the bottom left corner and the T-motor with the 16x5.5 propeller results are shown in the bottom right corner.

and inputting $V_\infty = 40$ m/s, $\rho = 0.1948$ kg/m³ taken from [26], $S = 1.296$ m², and $L = 68.74$ N, $C_L = 0.340$ is obtained. The calculated C_L gives an α angle in the 0° to 3° range from the CFD result table in the appendix. An α of 0° and 3° has $C_D = 0.0242$ and $C_D = 0.0277$ respectively. The higher C_D of the two is chosen for the calculations. Using the drag equation (15) for $C_D = 0.0277$ a drag force of $F_d = 5.59$ N = 570 gf is obtained. The drag force has to be matched by the thrust of all four motors of ALPHA to maintain speed. Thrust per motor comes out to $T_D = 142$ gf.

By applying equation (13) and equation (7) for all tested propellers, Table III is generated. By comparing the propeller RPM requirements in the table to the capability of the motors in Fig. 5, it is concluded that the 5x5, Gemfan 5040, Gemfan 7040 and 8x4.5 propellers are beyond the safe RPM range for any of the motors. The 12x4.5, 14x5.5 and 16x5.5 propellers are more than twice over their RPM limit, even exceeding the speed of sound at the wingtips. The only propeller rotating at a speed that keeps its wingtips significantly below the speed of sound while also remaining in the RPM range of some of the tested motors is the 10x7 propeller. This propeller is best suited because of its relatively high pitch, allowing a lower RPM. It would still be over its recommended 15000 RPM limit, although in [18] there exists APC test data up

to 24000 RPM which indicates that 17600 RPM would be feasible. Because of the very low required torque, only 0.0256 Nm, the six highest Kv-rated motors in Fig. 5 would all be capable of spinning the 10x7 at the speed and torque required to maintain a speed of 40 m/s at 15 km altitude. The capable motors are Leopard hobby 1600 Kv, Aerodrive 2836 1500 Kv, ZOHD 1300 Kv, Aerodrive 3548 1050 Kv, Emax Grand turbo 985 Kv and Sunnysky 920 Kv.

F. Sources of error

A number of error sources are identified in both test setups which might explain some of the inconsistencies seen in certain data points. As described in Section III propeller performance is dependent on air density in the testing environment. Air density is dependent on local temperature and air pressure which varies from day to day. However all test are performed in the same workshop over the course of two consecutive months so the impact of these factors is deemed insignificant. With the first setup using the Turnigy thrust stand and power analyser V3, it is impossible to perfectly set and read the PWM signal being sent. The signal is set by observing the PWM signal on an oscilloscope and setting it as precisely as possible. The data analysed is deemed sufficiently consistent

to be applicable, but reading errors are observed in multiple sets of data collected from the first setup.

With the second setup using the Tyto robotics RC Benchmark thrust stand, data collection is automatic using associated programmable software and PWM signals are exact, but some measurement errors are observed. Some files contain flawed RPM, torque or thrust values, that are inconsistent with the overall data or physically impossible. The Sunnysky 920 Kv motor test results for the 14x5.5 and 16x5.5 propellers are both useless due to an error that gave constant torque across the whole measurement. This happened in more cases as well, which is the reason some motors are missing certain propeller results. A large amount of data is collected using this setup and the effects of measurement error is deemed to be of low impact since the context of the data can be interpreted from the data as a whole. Another factor that might impact the results is the fact that the power supplies used to power the setup each generate around 12.2 volts with some variation. As power draw increased, voltage would drop by up to two volts. As earlier discussed, the voltage supplied directly affects the performance of the motor and ESCs, meaning the performance data cannot be guaranteed to be for 24 V voltage.

The continuous sweep used to gather data in the Tyto robotics 1585 tests increases the PWM signal from 1000 to 2000 μ s continuously, meaning the motors are accelerating throughout the tests. The acceleration of the propellers add to the measured torque, meaning the thrust to torque ratio is disturbed especially at low RPM. This also disturbs the propeller efficiency measurements since the mechanical power is internally calculated with equation (4) by RCbenchmark using 16. In reality however only the power resulting from drag on the propeller should be used in the calculation of the propellers efficiency, not the power accelerating the propeller. This could be solved with a discrete sweep on the Tyto robotics test stand where the propeller is not accelerating during measurements.

The approximation in equation (7) that torque is only linearly dependant on thrust is as mentioned a questionable approximation. This approximation only accounts for drag related to lift. This formula should be completed with a term for the drag resulting from the forward motion of the propeller blades through the air.

G. Future Work

To implement a propulsion system on the ALPHA UAV, the propulsion system should undergo further analysis before the required performance can be assured. Dynamic thrust of the propulsion system can be tested experimentally by placing the test setup in a wind tunnel and assessing the resulting performance. The results of such a study, could draw comparisons to the results discussed in this project and might change the requirements placed on the power electronics. Further analysis of the number of blades on the propeller is also required. The ALPHA UAV will be operating in a harsh environment with regards to temperature, which might affect the power electronics. According to [5] the temperature above eleven kilometres of altitude is negative 57 degrees Celsius

and the power electronics applicable for the ALPHA UAV should be tested at these temperatures to ensure the desired performance is retained. On the other hand both the electric motors and ESCs develop notable heat when operating and this factor should be included in the performance assessment for the selection process. Connecting a temperature probe to both the motor and ESC while performing tests similar to the ones conducted in this study, could yield essential information on the heat generated by the propulsion system. It is possible that this generated heat could be directed and used for ice and temperature protection for other part on the ALPHA UAV.

The method and results presented in this project lay the foundation for future work and studies on the ALPHA UAV. The final power electronics selected may differ from the ones studied in this project, but the performance can be compared to the performance parameters in this study and hopefully aid in motivating the selection. The results can also be compared to an analysis of flight data once test flights are completed and improvement can then be made to this study.

As a future addition to the motor tests, all motors should be tested without any propeller in order obtain the zero torque RPM for all motors across their throttle range. This data would close off the bottom of all plots in Fig. 5. Filling in this gap in the performance data would be especially useful for the high Kv motors where there is a large gap between the smallest propellers torque curve and the horizontal axis at high RPM. This area of the torque and RPM plots is very interesting for high speed low torque applications.

All propeller tests should be retested with a discrete sweep program on the Tyto-robotics 1585 thrust stand. As opposed to the continuous sweep done in this project, this would solve the problem of propeller momentum acceleration disturbing the torque data. With a discrete sweep the RPM would be constant during the data measurement. Plots of thrust and torque for propellers should, with this fix in place, begin in the origin and be nearly perfectly linear, erasing the initial jump in torque that results from the propeller accelerating quickly.

APC propellers provides a large selection of test data for their propellers in [18], including data for their performance at different velocities. This data is much more complete than that collected in this project. It would enable the creation of dynamic thrust plots for the propellers. This could aid the completion of a better model of equation (7) with the correction proposed.

VI. CONCLUSION

From the ESC tests it is seen that the type of ESC and the timing setting of these, does not play a significant role in the performance. The differences seen are insignificant enough to be attributed to errors. It should be mentioned that the test results show that the auto timing feature works optimally for the T-motor brand motors and an optimal fixed timing setting should be found and used for the Aerodrive brand motors. The current limitations of the tested ESC's were sufficient for accommodating the system for both 12 and 24 volt tests and when considering other ESC's that are commercially available, these limits should be taken into consideration as they might

impede the performance. If any conclusion should be drawn from the results attained in this project, it is that the brand of the ESC should match the brand of motor used. As seen in Fig. 8, bottom left plot, the T-motor brand ESC auto timing performs best with the T-motor brand motor. Apart from its interaction with the rest of the system, the T-motor brand ESC is advantageous due to its small size and low weight. This ESC also offers a wide range of settings that can be adjusted using interfacing software.

The tested motor deemed best suited based on Fig. 5 is Emax Grand turbo 985 Kv because of its wide performance range in regards to RPM and torque. This motor would have to be paired with the 10x7 propeller in order to have sufficient performance. In this combination at 15 km altitude and 40 m/s velocity, the motors throttle would be at around 1900 μ s. This is at the upper edge of the motors capability but some margin still remains.

The components tested in this project represent only a small selection of the available options for power electronics in this application. Based on the results, a propeller of high pitch with a diameter of around ten inches is recommended along with a light 24 V capable motor with a rating over 900 Kv. Propeller pitch should not be below seven inches and the pitch to diameter ratio should be high as to lower the required RPM and avoid high wingtip velocities.

APPENDIX

CORRECTED RESULTS SUMMARY

ACKNOWLEDGMENT

The authors would like to extend a big thank you to Nickolay Ivchenko for great supervision and guidance for this project and for creating a good atmosphere for learning and experimenting.

REFERENCES

- [1] M. Gaskill. (2018, Apr.) Nasa - once upon a time in a thunderstorm. NASA Johnson Space Center, Houston, TX, USA. [Online]. Available: https://www.nasa.gov/mission_pages/station/research/Once_Upon_a_Time_in_a_Thunderstorm
- [2] (2022, Apr.) Space weather glossary. Space Weather Prediction Center, Boulder, CO, USA. [Online]. Available: <https://www.swpc.noaa.gov/content/space-weather-glossary>
- [3] European Space Agency. (2017, Feb.) Esa - blue jets studied from space station. ESA HQ Bertrand, Paris, France. [Online]. Available: https://www.esa.int/Science_Exploration/Human_and_Robotic_Exploration/iriss/Blue_jets_studied_from_Space_Station
- [4] KTH Royal Institute of Technology. (2022, Apr.) Alpha — kthaero. KTH Aerospace, Stockholm, Sweden. [Online]. Available: <https://www.kthaero.com/alpha>
- [5] CAE, *Meteorology ATPL Ground Training Series*. London, United Kingdom: CAE Oxford Aviation Academy (Oxford) Ltd., 2018, chapter 12, pp. 197 and chapter 11, pp. 180.
- [6] Airbus. (2021, Aug.) Zephyr. Airbus Group SE, Leiden, Netherlands. [Online]. Available: <https://www.airbus.com/en/products-services/defence/uas/uas-solutions/zephyr>
- [7] (2022, Apr.) Boeing: Phantom eye. The Boeing Company, Illinois, IL, USA. [Online]. Available: <https://www.boeing.com/defense/phantom-eye/>
- [8] D. L. Gabriel, J. Meyer, and F. du Plessis, "Brushless dc motor characterisation and selection for a fixed wing uav," in *IEEE Africon '11*, 2011, pp. 1–6.
- [9] C. R. Green and R. A. McDonald, Eds., *Modeling and Test of the Efficiency of Electronic Speed Controllers for Brushless DC Motors*. American Institute of Aeronautics and Astronautics, Inc., Jun. 2015.
- [10] A. Hughes, *Electric Motors and Drives, Third edition*. Burlington, MA, United Kingdom: Elsevier Ltd., 2006, chapter 5, pp. 167-183 and chapter 2, pp. 65-69.
- [11] D. Nedelkovski. (2019, Feb.) How to mechatronics - how brushless motor and esc work. Amazon Inc., Washington, WA, USA. [Online]. Available: <https://howtomechatronics.com/how-it-works/how-brushless-motor-and-esc-work/>
- [12] J. Reid. (2017, Jan.) Rotor drone - understanding kv ratings. Air Age Media, New York, NY, USA. [Online]. Available: <https://www.rotordronepro.com/understanding-kv-ratings/>
- [13] D. Nedelkovski. (2019, Feb.) How to mechatronics - arduino brushless motor control tutorial. Amazon Inc., Washington, WA, USA. [Online]. Available: <https://howtomechatronics.com/tutorials/arduino/arduino-brushless-motor-control-tutorial-esc-bldc/>
- [14] (2019, May) Timing explained for brushless motor and esc's (low vs high). Amazon Inc., Washington, WA, USA. [Online]. Available: <https://www.radiocontrolinfo.com/timing-explained-for-brushless-motor-and-escs/>
- [15] O. Liang. (2017, Feb.) Oscar liang - motor timing. Cloudflare Inc., California, CA, USA. [Online]. Available: <https://oscarliang.com/motor-timing/>
- [16] J. Anderson, *Fundamentals of Aerodynamics, Sixth Edition*. New York, NY: McGraw-Hill Education, 2017, chapter 1, section 1.14, pp. 93.
- [17] Massachusetts Institute of Technology. (2008, Sep.) Performance of propellers. MIT, Massachusetts, MA, USA. [Online]. Available: <https://web.mit.edu/16.unified/www/FALL/thermodynamics/notes/node86.html#SECTION06374200000000000000>
- [18] APC Propellers. (2021, Mar.) Performance data. Landing Products Inc., California, CA, USA. [Online]. Available: <https://www.apcprop.com/technical-information/performance-data/>
- [19] CAE, *Principles of Flight ATPL Ground Training Series*. London, United Kingdom: CAE Oxford Aviation Academy (Oxford) Ltd., 2018, chapter 15, pp. 491.
- [20] (2014, Jan.) Dynamic thrust. Wordpress, California, CA, USA. [Online]. Available: <https://itsallrc.wordpress.com/2014/01/06/dynamic-thrust/>
- [21] (2014, Apr.) Propeller static dynamic thrust calculation - part 2 of 2 - how did i come up with this equation? Blogspot, California, CA, USA. [Online]. Available: <https://www.electricrcaircraftguy.com/2014/04/propeller-static-dynamic-thrust-equation-background.html>
- [22] APC Propellers. (2022, Apr.) Rpm limits. Landing Products Inc., California, CA, USA. [Online]. Available: <https://www.apcprop.com/technical-information/rpm-limits/>
- [23] APC Propellers. (2020, Aug.) Manufacturing. Landing Products Inc., California, CA, USA. [Online]. Available: <https://www.apcprop.com/technical-information/manufacturing/>
- [24] (2022, Apr.) Hobbyking - gemfan bullnose polycarbonate 5040 6-bladed propeller. Hobbyking, Kwun Tong, Hong Kong. [Online]. Available: https://hobbyking.com/en_us/5040-pc-6-blade-black.html?store=en_us
- [25] (2022, Apr.) Fpv24 - gemfan 7040 flash 3 blade propeller. Meilon GmbH, Remagen, Germany. [Online]. Available: <https://www.fpv24.com/en/gemfan/gemfan-7040-flash-3-blatt-propeller-schwarz-4-stueck>
- [26] Engineering ToolBox. (2003, Jun.) U.S. Standard Atmosphere VS Altitude. [Online]. Available: https://www.engineeringtoolbox.com/standard-atmosphere-d_604.html

CONTEXT M

ARTIFICIAL INTELLIGENCE FOR THE INTERNET OF THINGS

POPULAR DESCRIPTION

Smart utopia or Terminator apocalypse?

Checking in on our babies over a wireless baby monitor might be convenient, but how do we make sure we are the only ones watching? Teaching a machine to detect poisoned water might sound like a no-brainer, but what's to stop the same technology from being used in terrorism? Toasters connected to fridges, cameras connected to servers on a different continent, and all of these devices are constantly pumping out massive amounts of data. The questions "This is amazing, what can we do with it?" and "This is amazing, but how do we keep it secure?" need to be seen as equals.

This new level of digitalization is creating what is called an Internet of Things (IoT), from the office at home, to global cooperation. A common saying in the IT world is that the "S" in IoT stands for security, highlighting the absence of a security focus in IoT. Security, integrity and society's well-being is often a secondary priority compared to being first with the next big thing in IT. Connecting everything and teaching machines to handle all this data productively is a massive opportunity in everything from smart homes to environmental sciences. However, whether that future world currently only seen in science fiction is an utopian Star Trek or that of Arnold Schwarzenegger's Terminator dystopia is still to be decided.

How can we teach machines to do amazing things, while having many computers share the workload? Can we teach them to prevent buffering on streaming services? How can we use a bunch of sensors to monitor the quality of drinking water? Can we demonstrate the security issues of IoT devices by hacking one? How can we predict cyberattacks and help its users in keeping their systems secure? These are some of the issues that must be answered when we move into this new era, a small part in ensuring this new revolution is of benefit to humanity and not its downfall.

SUMMARY OF PROJECT RESULTS

An increasingly digitized and connected world brings with it both opportunities and challenges. Modern networks are complex, large and can allow any previously analog object to actually make decisions. The onset of this Internet of Things (IoT) brings with it an ability to analyze extensive datasets, reaching previously unachievable results using machine learning (ML) and other methods.

However, new developments in computer networking also brings with it concerns in the areas of security and privacy. Larger and more complex networks and the desire to intensely digitalize the world around us also results in larger attack surfaces, more devices to hack, and critical data potentially reaching a malicious observer. Context M deals with both reaping the rewards and ensuring the security of these new systems.

The advances in the field of ML have given powerful tools for estimating service metrics within networks. However, monitoring a network to create a large enough data set is costly, not mentioning the required cache sizes to store said data and the computational power to process it. An approach to make predictions using ML within resource constrained environments is "Online Learning," where fewer samples are used. With this, group M4 has found new approaches for building smaller datasets used to train models predicting service quality within networks, achieving similar accuracy as using

large datasets, but with greatly reduced overhead. Predicting service quality will become increasingly important as networks grow in complexity with further additions, such as IoT.

Using IoT devices to do ML calculations can cause new issues. The project M1 aims to analyze the process of performing calculations over large networks in a more theoretical sense, looking at the current scientific landscape in distributed ML and examining latency and performance of distributed ML through simulations. The project sets the stage for more practical uses of ML through distributed IoT networks. While the influence of ML grows, so does the responsibility to develop such powerful tools. Therefore, group M1 focused on building a Deep Neural Network from the ground up, in order to understand how common ML algorithms work on a deeper level.

Drinking water's cleanliness is a constant concern, and the identification of sewage in it is extremely important. Currently, the most common method of identification is manual data processing. Group M2 studied data received from sensors from two different locations in Linköping in order to identify potential contamination in a testing site measuring different qualities in drinking water. The sensors measured different parameters ranging from temperature to the level of chlorine inside the water. The method of identification had to be self-sustainable and with as few false alarms as possible. This was to be implemented using a Deep Neural Network and Machine Learning. A clear indication between some features and contamination were identified. This will provide the analysis of data provided from the sensors into the IoT in future projects.

Many everyday devices such as kitchen gadgets, baby cameras, vacuum cleaners have in recent years been connected to the IoT. As devices become connected, it becomes increasingly relevant that they are secure. The aim of project group M5 was to evaluate the security of an IoT device. The project group decided on evaluating an IP and baby camera available at a large retailer in Sweden. The camera was found to be lacking in regard to cybersecurity, risking a breach of privacy for the consumer. The results of the project sheds light on the importance of cybersecurity in regard to the many IoT-devices that are being developed today.

Project M6 deals with tackling the issue of cybersecurity in modern networks in a more general manner, requiring the ability to assess and analyze the security of different domains in a structured and formalized way. The Meta Attack Language (MAL) was recently developed by researchers at KTH in an attempt to provide a formal way to describe systems using graphs. These graphs can then be used to simulate attacks on that system and assess its security risks. The aim with project M6 was to create a proof of concept that the results of the attack simulations can be used to automate actual penetration tests on the environment. Future improvements include extending the functionality to create a more fully-fledged penetration testing tool based on MAL.

Future projects in this context should consider the effectiveness and security of new systems as two sides of the same coin. Projects M4 and M6 deal with cybersecurity as its own isolated problem, while projects M2 and M4 deal with utilizing new opportunities that modern networks provide. New opportunities and new risks are handled as separate issues. Ensuring the security of distributed ML and big data analysis should be one of its core requirements, on the same level as its results and efficiency.

IMPACT ON SOCIETY AND ENVIRONMENT

When analyzing the impact of a society that is increasingly connected, it is relevant to examine the effects on individuals, society as a whole and the environment. Digitalizing our homes, offices, cities and energy systems is often seen as a clear positive development for society. It is however necessary to analyze the effects of these developments at every level as well as consider how and if advancements should be made, and how we can ensure that the development is sustainable and not a part of a mindless drive for digitalization for the sake of digitalization.

The interconnected world of IoT has a potentially massive impact on an individual level. Everyday life can be improved by simplifying mundane tasks through smart homes and offices, providing health benefits through medical IoT and improving accessibility in digital communications and off-site offices. However, digitalizing modern living also brings with it risks in security and integrity. A connected individual is an exposed individual, and collecting large amounts of personal data for ML

algorithms often entails disregarding consent and inadequate transparency. Combining a digitalized life and a profit-motive with the right to personal integrity and freedom often carry risks for the individual; whether through data leaks, malicious use or hacking.

Conceivable negative effects on the individual can however be positive for society as a whole. For example, camera surveillance and similar forms of pre-emptive data collection for law enforcement can be beneficial for policing and security. While this offers new opportunities in society, it is crucial that it is used and developed with integrity and concepts such as the right to be considered innocent until proven guilty in mind. A government collecting massive amounts of data on its citizens might be useful, for example to tailor the availability of the public transport system. However, there are certain unalienable rights and problems that have to be considered, even disregarding the issue of security and data falling into the wrong hands.

In regard to ML being used on the societal level, it is necessary to be aware of the risks of biased algorithms. Biased algorithms could affect different groups in society unevenly, for example an ML algorithm developed by Amazon to be used to hire engineers was trained on data which was unintentionally biased towards hiring men, rendering the algorithm unusable. Another potential issue is regarding the availability of newly developed IoT products and networks. Differences in technological prowess and awareness within different groups, combined with economic conditions, could create a strongly divided society where some do not have access to the technology that will shape our society in the future.

The development of decentralized and smart sensors could allow complex environmental questions to be tackled by analyzing large amounts of previously inaccessible data. Smart sensors may enable heavier and more stringent data collection, which may be processed with the use of ML to provide solutions for environmental issues. However, widespread use of these new IoT networks in environmental sciences can also in themselves have a negative impact by creating a greater demand on natural resources from manufacturing and development. Additionally, new technology and products might be short-lived and quickly become obsolete, producing more electronic waste. Improving efficiency could on a larger scale reduce the environmental impact, but could also lead to over-engineering environmental issues. Attempts to find complicated technological solutions to non-technological problems could paradoxically result in an overall greater consumption of natural resources.

At every level, digitalization with IoT networks and collecting data for ML have conflicting results that need to be balanced. On the individual level, convenience has to be balanced with integrity. On the societal level, efficiency has to be balanced with security and personal rights. Finally, on the environmental level, using sensors, algorithms and ML has to be done with care and when needed, not as a catch-all solution for climate change or pollution. Digitalization will bring huge benefits on every level of the human experience and the world around us, but it can also carry risks of the same magnitude if not done with care. Connecting our world in large, distributed networks that constantly collect massive amounts of data should be done when the result is a net benefit examined on a case-by-case basis, not as advancement for the sake of advancement.

Building and Training a Fully Connected Deep Neural Network From Scratch

Axel Berglund

Abstract—Artificial Neural Networks make up the core of most Machine Learning algorithms. In the past decade Machine learning have successfully taken on fields such as image recognition, Data analytics and medical technologies. As the area of use become less prone to mistakes, it raises the responsibility look into the black box of code and understand it to a deeper level. In this project, I built a Deep Neural Network from scratch, without high level libraries, and trained it for a supervised classification task. The finished algorithm is flexible and can be adapted to any classification problem. The training method is based on Backpropagation and Gradient Descent. At last, the algorithm was trained on the Modified National Institute of Standards and Technology (MNIST) database, and performed with a 77% prediction accuracy. There are a few optimization methods yet to be tested to further increase the performance.

Sammanfattning—Artificiella neurala nätverk utgör kärnan i de flesta maskininlärningsalgoritmer idag. Under det senaste decenniet har maskininläring framgångsrikt tagit an områden som bildigenkänning, dataanalys och medicinsk teknik. När användningsområdena blir mindre benägna till misstag, ökar ansvaret av att titta under huven och förstå den djupare nivåkoderna. I denna studie var syftet att bygga ett djupt neuralt nätverk från grunden, utan högnivåbibliotek, och träna det för en övervakad klassificeringsuppgift. Den färdiga algoritmen är flexibel och kan designas för flera klassificeringsproblem. Nätverkets träningsmetod är baserad på Backpropagation och Gradient Descent. Valideringsdatan kunde till slut köras med 77% korrekt noggrannhet, och det finns ytterligare optimeringsmetoder att testa för att höja prestationen.

Index Terms—Deep Neural Network, Machine Learning, Gradient Descent, MNIST.

Supervisors: Henrik Hellström

TRITA number: TRITA-EECS-EX-2022:162

I. INTRODUCTION

Neural Networks (NN) have played an important role for the progress of pattern recognition systems. NNs were first proposed in 1944 and have had a tough developing journey. However, as larger labeled datasets became available, together with faster processors, the power of Deep Neural Networks (DNN) began to thrive around 2010 [1] [2]. Today DNNs can, with the right architecture, find high dimensional patterns in any labeled dataset. Image recognition, speech recognition and medicine all use Machine Learning (ML) algorithms with great success [3]. The challenge with ML algorithms, particularly with NNs, is to find the right architecture and train it optimally. It may be time consuming to train the program but once it is trained, it can operate with low computational complexity [4]. The aim of this project is to write code representing a DNN

and train it for a supervised classification task. No high level libraries will be used other than linear algebra library numpy. The target goal is to break 50% accuracy on MNIST dataset. Performance will be limited as the number of training epochs are restricted. The program is built to work for any classification problem, but the presented measurements come from one particular problem: to classify digits in images. Different architectural structures have not been tested in comparison to the one used, which leaves room for further testing and possible improvement.

II. SYSTEM MODEL

A. Fully-Connected Neural Network

A Neuron consist of one weighted link, one bias term and an activation function. It takes a scalar x as input which is multiplied with a weight w , and then a bias term b is added which returns z . The neuron output a is then computed by running the resulting value through an activation function.

$$z = wx + b \quad (1)$$

$$a = \sigma(z) = \frac{1}{1 + e^{-z}} \quad (2)$$

A DNN is built with a large numbers of Neurons. Especially layers of neurons. For a NN to be called DNN, it must contain at least three layers. One input layer, one Hidden layer and one output layer. It is the hidden layer that creates the abstract representations of the data. Pattern recognition complexity increase by adding more hidden layers to the network. The built network for this project is demonstrated in figure 1. Each neuron in one layer is connected to every neuron in the next layer. It is therefore convenient to store all weighted links of layer L in a matrix $\mathbf{W}^{[L]}$:

$$\mathbf{W}^{[L]} = \begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} & \dots & w_{1,n} \\ w_{2,1} & w_{2,2} & w_{2,3} & \dots & w_{2,n} \\ w_{3,1} & w_{3,2} & w_{3,3} & \dots & w_{3,n} \\ \dots & \dots & \dots & \dots & \dots \\ w_{N,1} & w_{N,2} & w_{N,4} & \dots & w_{N,n} \end{bmatrix} \quad (3)$$

Where n is the number of neurons in layer L-1, and N is the number of neurons in layer L. The output from the first layer becomes the input to the next layer. The total sum going into each neuron one layer is represented by vector \mathbf{z} :

$$\mathbf{z}^{[L]} = \mathbf{W}^{[L-1]} \mathbf{a}^{[L-1]} + \mathbf{b}^{[L-1]} \quad (4)$$

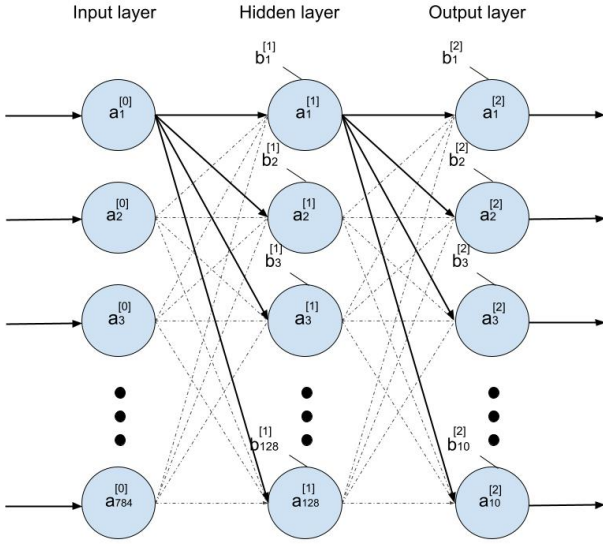


Fig. 1. The built NN structure with number of layer units presented

$$\mathbf{z} = \begin{bmatrix} w_{1,1}a_1 + a_2w_{1,2} + a_3w_{1,3} + \dots + a_nw_{1,n} + b_1 \\ w_{2,1}a_1 + a_2w_{2,2} + a_3w_{2,3} + \dots + a_Nw_{2,n} + b_2 \\ w_{3,1}a_1 + a_2w_{3,2} + a_3w_{3,3} + \dots + a_Nw_{3,n} + b_3 \\ \dots \\ w_{N,1}a_1 + a_2w_{N,2} + a_3w_{N,3} + \dots + a_nw_{N,n} + b_N \end{bmatrix} \quad (5)$$

Resulting input to each layer is therefore a vector \mathbf{z} as shown in 5. The dimension of the weight matrix matches the number of units in its surrounding layers. Bias vectors \mathbf{b} are of same length as its associated layer.

B. Inference

During inference, data passes forward through all layers and infers a result. Each layer processes the information per the activation function, and it becomes the input for successive layer. This method is called forward propagation [5]. All transitional variables between the layers are temporarily stored for the purpose of training the algorithm. Neuron activation functions may vary depending on the problem. I present two different activation functions in this project, the Sigmoid- and Softmax function. The hidden layer uses the Sigmoid function presented in (2), and the Softmax function is applied on the output layer.

When testing the performance, onehot encoding is used to classify the output probabilities as binary. The output vector is representing probability for each digit and the onehot method encodes largest value as one and set the rest to zero.

C. Training

In order to train the program we must keep track of how well the algorithm is performing. Therefore, we implement a loss function, also known as a cost function, to calculate the error. I have used the Mean Squared Error (MSE) as the loss function and its purpose is to encode how "bad" the neural network

is at classification (i.e., if the prediction deviates much from the actual values, the loss function will return a large value). The idea is to train the neural network by reducing the error after each iteration until the minimum to the loss function is located. This function stores the error, the cost, to the predicted output the true output, and stores the error.

$$J = MSE = \frac{1}{M} \sum_{m=1}^M (a^{[2]} - Y)^2 \quad (6)$$

The error for each epoch is stored and the sum is divided by the number total number of iterations, described in (6) as M . The goal is to minimize the calculated loss, by tuning the weights and bias terms between every epoch. I have used Gradient Descent (GD) as a learning method to minimize the loss function. The loss function gradient tells us which direction the loss is increased, with respect to a certain parameter. Loss can therefore be decreased by moving in the direction opposite to the gradient's [6]. GD is an optimization algorithm used for its computational efficiency, but in order to perform an update, the algorithm must be able to calculate a gradient of the loss function with respect to the weights and biases. For a deep neural network, direct computation of the gradient is computationally complex, which nullifies the advantage of using GD. Therefore, an efficient method for calculating the gradient is required. One such method is backpropagation. Since the feed forward phase stores almost all transitional variables, we can use them to calculate the gradients backwards through the network. This method is called backpropagation, and we can use it to update the parameters before going to the next epoch.

$$\theta_{i+1} = \theta_i - \alpha \frac{\partial}{\partial \theta_i} J(\theta) \quad (7)$$



Fig. 2. Data samples from MNIST

Here θ is one of the parameters and i represents an iteration. α is the step size: what scale the parameter should be updated with. One way to optimize the algorithm performance is to find the optimal step size. Since if the step size is too small, it could

TABLE I
MNIST DATA CLARIFICATION

Dataset	Dataset length	Dimension
X_{train}	60000	(784,1)
X_{test}	10000	(784,1)
Y_{train}	60000	(10,1)
Y_{test}	10000	(10,1)

take too long to reach the minimum of the loss function. Or if step size is too large it may overshoot the minimum instead. Another way to optimize the algorithm performance is to use Batch Gradient Descent (BGD). Which means that instead of updating the parameters after each single data point, one could create a batch of data points and only take an update once the gradient for all data points in the batch have been calculated. Therefore the batch size is used as a hyperparameter. There are two different types of parameters that are adjusted when training the NN to affect the loss function: weights and biases. The partial derivative of the cost function, for each individual parameter, can be calculated using the chain rule, with the help of backpropagation. The whole algorithm is summarized in Algorithm 1.

$$\frac{\partial C}{\partial W^{[L]}} = \frac{\partial C}{\partial a^{[L]}} \frac{\partial a^{[L]}}{\partial z^{[L]}} \frac{\partial z^{[L]}}{\partial w^{[L]}} \quad (8)$$

$$\frac{\partial C}{\partial b^{[L]}} = \frac{\partial C}{\partial a^{[L]}} \frac{\partial a^{[L]}}{\partial z^{[L]}} \frac{\partial z^{[L]}}{\partial b^{[L]}} \quad (9)$$

$$\frac{\partial C}{\partial W^{[L-1]}} = \frac{\partial C}{\partial a^{[L]}} \frac{\partial a^{[L]}}{\partial z^{[L]}} \frac{\partial z^{[L]}}{\partial a^{[L-1]}} \frac{\partial a^{[L-1]}}{\partial z^{[L-1]}} \frac{\partial z^{[L-1]}}{\partial w^{[L-1]}} \quad (10)$$

$$\frac{\partial C}{\partial b^{[L-1]}} = \frac{\partial C}{\partial a^{[L]}} \frac{\partial a^{[L]}}{\partial z^{[L]}} \frac{\partial z^{[L]}}{\partial a^{[L-1]}} \frac{\partial a^{[L-1]}}{\partial z^{[L-1]}} \frac{\partial z^{[L-1]}}{\partial b^{[L-1]}} \quad (11)$$

Note that these partial derivatives are with respect to single weights, and single bias terms. Every weight and bias is updated using (7).

III. SIMULATION

A. MNIST

MNIST is a commonly used dataset which consist of 70000 images of handwritten digits. See figure 2. Each image is labeled with what number the image represents. The data is divided into one pile of training data, and one pile of validation data, with 60000 and 10000 data respectively. Each image contains 28x28 pixels with a grayscale value ranging from zero to 255. The task is to study an image and classify what digit it represents.

B. Neural Network parameters

The input vector $\mathbf{a}^{[0]}$ is fetched by flattening the image into one column with 784 pixels:

$$\mathbf{a}^{[0]} = [a_1 \ a_2 \ a_3 \ \dots \ a_{784}]^T \quad (12)$$

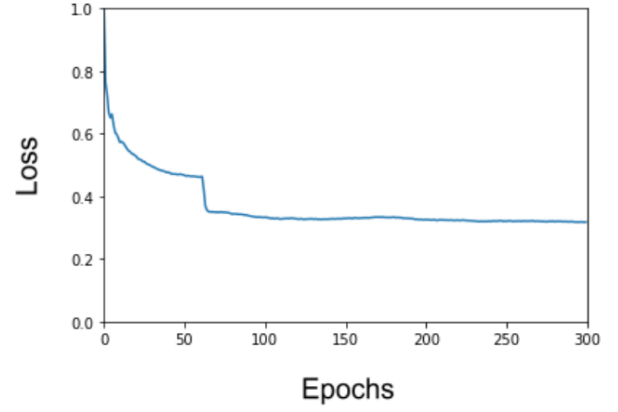


Fig. 3. Loss function plotted over number of epochs. The sudden drop around 60 epochs is thought to be because 8s are starting to be recognized

The number of units in the hidden layer was set to 128. The output layer is of dimension ten (i.e., one node for each digit). The three layers are linked by two weight matrices, of dimensions proportional to the layer dimensions:

$$\mathbf{W}^{[1]} \in R^{128 \times 784} \quad \& \quad \mathbf{W}^{[2]} \in R^{10 \times 128}$$

The step size was set to $\alpha = 0.01$. The maximum number of epochs tested was 300.

Algorithm 1

Require: *batchSize*

```

for i in range epochs do
    gradients = 0
    batch = 0
    for X in  $X_{train}$  do
         $Y_{pred} = forwardPass(X) \leftarrow Equation(4)$ 
         $Loss = MSE(Y_{pred} - Y) \leftarrow Equation(6)$ 
         $gradient += backpass(Loss)$ 
     $\leftarrow Equation(8)(9)(10)(11)$ 
    if batch = batchSize then
        Update(gradients)  $\triangleright$  Summed batch update
        batch = 0
        gradients = 0
    else
        batch += 1
    end if
end for
correct = 0
wrong = 0
for X in  $X_{test}$  do
     $Y_{pred} = forwardPass(X) \leftarrow Equation(4)$ 
     $Y_{pred} = OneHot(Y_{pred})$ 
    if  $Y_{pred} = Y$  then
        correct += 1
    else
        wrong += 1
    end if
end for

```

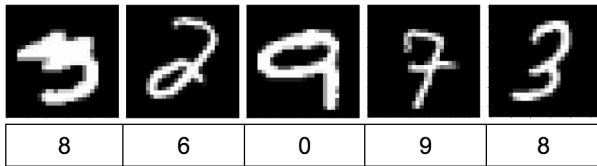


Fig. 4. These images were wrongly classified and the table shows the predictions. The predictions are wrong but not always bad

C. Results

The algorithm was tested on the remaining 10000 test images and performed with 77 % accuracy at best. Different parameter values were tested, such as stepsize, batchsize and the number of epochs. The Loss function curve was plotted over number of epochs and it seems to yet be decreasing slowly. Future work on this project ought to consider training the algorithm with larger number of epochs.

IV. DISCUSSION

As shown in figure 3, the loss function value seems to be slowly decreasing even after 300 epochs. It may be so that the minimum point has not been reached yet. What also is interesting is the sudden curve drop around 60 epochs. After investigation it was thought that the drop can be explained by the algorithms ability to classify the digit 8. Before the drop, the algorithm was never able to classify digits 8 or 1. After the drop, 8s started to appear among the correct predictions. Leaving the 1s to be the only unsuccessfully predicted digit. There are ways to try and optimize the program for higher accuracy. By increasing the step size one might find a minimum point in the loss function at last. Current step size was settled by tuning it until the prediction accuracy reached its peak. Another method to consider implementing is dynamic step size.

I found it particularly interesting that even if a prediction is wrong, it can still be reasonable. In figure 4 one can see how some digit predictions resemble the true digit. Some images are drawn sloppy and may even be hard for a human to classify.

V. CONCLUSION

The created NN is flexible and can be applied for any classification task in supervised ML. The user of the program can choose the step size, batch size and the number of neurons in the hidden layer. I optimized the algorithm and exceeded the goal by predicting output with 77% accuracy. The results show that the NN learn from the training process. Images show that the network troubles with identifying 8s and 1s. After 60 epochs it learns to separate 8s from other digits.

APPENDIX A DEEP NEURAL NETWORK ALGORITHM

ACKNOWLEDGMENT

I am sincerely grateful for all the support from supervisor Henrik Hellström. Every meeting was highly productive, and Hellström's natural teaching skills kept me motivated to learn.

REFERENCES

- [1] R. Salakhutdinov and G. Hinton, "An Efficient Learning Procedure for Deep Boltzmann Machines," *Neural Computation*, vol. 24, no. 8, pp. 1967–2006, 08 2012. [Online]. Available: https://doi.org/10.1162/NECO_a_00311
- [2] R. Raina, A. Madhavan, and A. Ng, "Large-scale deep unsupervised learning using graphics processors," in *Proceedings of the 26th Annual International Conference on machine learning*, ser. ICML '09. ACM, 2009, pp. 873–880.
- [3] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3642–3649.
- [4] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, "Advances in optimizing recurrent networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 8624–8628.
- [5] A.Zell, T.Korb, T.Sommer, and R.Bayer, "Neural network simulation environment," in *Applications of Artificial Neural Networks*, S. Rogers, Ed., vol. 1294, International Society for Optics and Photonics. SPIE, 1990, pp. 535 – 544. [Online]. Available: <https://doi-org.focus.lib.kth.se/10.1117/12.21204/>
- [6] S. Bubeck, *Convex optimization: algorithms and complexity*, ser. Foundations and trends in machine learning. Hanover, Massachusetts: Now Publishers, 2015, vol. 8, no. 3-4.

Water Contamination Detection With Binary Classification Using Artificial Neural Networks

Nicholas von Butovitsch and Christoffer Lundholm

Abstract—Water contamination is a major source of disease around the world. Therefore, the reliable monitoring of harmful contamination in water distribution networks requires considerable effort and attention. It is a vital necessity to possess a reliable monitoring system in order to detect harmful contamination in water distribution networks. To measure the potential contamination, a new sensor called an 'electric tongue' was developed in Linköpings University. It was created for the purpose of measuring various features of the water reliably. This project has developed a supervised machine learning algorithm that uses an artificial neural network for the detection of anomalies in the system. The algorithm can detect anomalies with an accuracy of around 99.98% based on the data that was available. This was achieved through a binary classifier, which reconstructs a vector and compares it to the expected outcome. Despite the limitations of the problem and the system's capabilities, binary classification is a potential solution to this problem.

Sammanfattning—Vatten kontaminering är en huvudsaklig anledning till sjukdom runt om i världen. Därför är det en avgörande nödvändighet att ha ett tillförlitligt övervakningssystem för att upptäcka skadliga föroreningar i vattendistributionsnät. För att mäta den potentiella föroreningen skapades en ny sensor, den så kallade "Electric Tongue" vid Linköpings universitet. Den skapades i syfte att mäta olika egenskaper i vattnet på ett tillförlitligt sätt. Genom att använda ett artificiellt neuralt nätverk utvecklades en supervised machine learning algoritmen för att upptäcka anomalier i systemet. Algoritmen kan upptäcka anomalier med 99.98% säkerhet som baseras på befintliga data. Detta uppnåddes genom att rekonstruera en vektor och jämföra det med det förväntade resultatet genom att använda en binär klassificerare. Trots att det finns begränsningar som orsakats både av problemet men också systemets förmågor, så är binär klassificering en potentiell lösning till detta problem.

Index Terms—Water distribution network, Machine Learning, Artificial Neural Network, Supervised Machine Learning, Binary Classification, Time Window

Supervisors: Henrik Hellström

TRITA number: TRITA-EECS-EX-2022:163

I. INTRODUCTION

Clean drinking water is a fundamental necessity to the health and survival of human beings. In modern times, drinking water is delivered to homes via pipes in large complex systems; however, they are susceptible to leakages, contaminating the drinking water. In the past, contamination in the water pipes may have been identified only through sickness rising in affected areas. Even currently, thousands of people in the USA die every year from unclean drinking water and contamination has even contributed to the illnesses of

millions [1]. The importance in identifying these contaminants is therefore very relevant and a prevailing problem even today.

However, in the Information Age there has also been a drastic increase and implementation of sensor technology, systems, and other related technologies (not in the least Machine Learning) culminating to a concept referred to as the Internet of Things (IoT).

IoT is a network of interconnected technologies which are able to exchange and broadcast data over the Internet and other communication networks [2]. Sensors are an integral part of IoT and can be used to collect large amounts of data. This data can be used in order to solve problems using machine learning. In the scope of this problem, large amounts of data are transferred from a sensor and recorded. This data can be used to detect the presence of contamination in drinking water.

The project's aim is to create an algorithm which, by using data received from sensors located in water treatment plants Nykvarn and Lingham, will be able to detect anomalies in the data and identify if there is a possible contamination in the water. The produced algorithm must have high accuracy in identifying contamination from otherwise non-contaminated data, as well as avoid missing said contamination. With improvements to sensors, computational power and theory, these incidents can be reduced significantly through advancements in the creation of effective algorithms.

This project report will outline a possible solution that uses supervised learning as well as a deep neural network. In Section II the problem is presented. In Section III the theory essential for full comprehension will be described. Section IV will walk through the decisions and thought processes that led to the project's result. Section V will present the project's result. Finally, Section VI will discuss certain aspects of the study; what was done and what could have been done differently.

The data for this project was provided Linköpings Universitet and created by a newly developed IoT sensor in Linköping for the detection of pollutants in drinking water called an 'Electric Tongue' [3]. According to the report, the sensor was shown to have significantly better results in the detection of contamination in drinking water in comparison to other sensors tested in the study. The sensors are dependent on time-related factors such as temperature, chloride concentration, water pressure, etc. These lead to risks and may cause false alarms in the system. It is far more

difficult to achieve a good result outside of lab conditions, where all parameters are controlled. However, in real world environments it is difficult to know and often rare for contamination to have occurred, making the process of testing for contamination very complex.

This report uses the following notation. Matrices and vectors are denoted as bold letters, e.g., \mathbf{A} . The transpose of a matrix or vector is denoted with a \top , e.g., \mathbf{A}^\top would be the transpose of \mathbf{A} .

II. PROBLEM FORMULATION

This project has the goal of creating a program with the intention of detecting anomalies in the water quality that are potentially caused by contamination. This program, using a machine learning algorithm on an ANN, should be able to detect such anomalies on the data provided by Linköpings Universitet. The data-sets contains both measurements by the *electric tongue* in a controlled environment as well as from real-world sites. In this project, a machine learning algorithm and a neural network structure will be used to solve this problem. Specifically, a binary classification neural network will be constructed, trained and evaluated upon using the data provided.

In order to achieve our central goal, a part of the project revolves around reducing the influence of time dependency in the neural network. This will eliminate factors that may not be relevant for the anomalies in the data-set as well as optimize the system. Another part of the project requires the neural network to observe time dependency in order to realize contaminants.

III. THEORY

A. Machine learning

It uses an algorithm with the purpose to optimize the prediction of an outcome without explicitly telling the algorithm to do so. Deep learning refers to a more specific method within ML which assimilates neural networks in several succeeding layers to learn from data with an iterative technique. Deep learning uses sample data known as training data and inserts it into an algorithm which predicts an output based on its training data. The following parts of this section will help build a fundamental understanding of artificial neural networks, stochastic gradient descent and pre-processing of data.

There are several ML techniques; supervised, unsupervised, semi-supervised, and reinforced learning. The choice of method largely depends on the application. Supervised Learning is a machine learning paradigm used with prior knowledge of the input and output from the sample data, known as the *training data-set*. The output is regarded as the label of the input. With supervised learning an input-output relationship should be created to correctly identify the label of the output. The goal of supervised learning is to build an artificial system that can learn the mapping between the

input and the output, and can predict the output of the system given new inputs. The goal is to correctly identify the label of the output and predict the label from a new data-set often known as the *testing data-set* [4]. For example, with the goal of developing an algorithm that can differentiate between a cat and dog, the algorithm will be fed with labelled data. The algorithm will optimize itself based on the difference between the hypothesis and the correct answer.

In contrast unsupervised machine learning is used to learn from unlabelled data. Unsupervised learning uses a data-set and finds structures such as grouping or clustering of unlabelled data-sets [5]. Unsupervised machine learning finds unknown patterns in data, for example, a man's identification of different breeds of dogs from the viewing of a single dog is a real-life example of unsupervised machine learning.

In this project supervised learning is the approach used.

1) *Neural Networks*: The neural network is a mathematical model that when iterated through, will produce a hypothesis, which is a numerical value. The hypothesis will vary depending on what the neural network is trying to do. In this case the neural network will output a zero or a one as it should inform whether the water is contaminated (1) or not (0).

There are three fundamental components of an ANN. First, there is the input layer where the neural network receives the information it needs to produce a hypothesis. Second, there are the hidden layers which hold most of the responsibility in calculating and coming up with the hypothesis. Hidden layers are a term used as they occupy layers in between the input and output layers and can be seen as private to the neural network. The number of hidden layers depends on the needs of the neural network; the higher quantity of layers there are, the more complex the neural network will become. Lastly, is the output layer where the results exit the algorithm. Each neuron will connect to another neuron and this connection will hold a certain 'weight'. Weights are numerical values that show the significance of the information that is being connected. The bias is an entirely separate value that is connected to each of the neurons in the hidden layer to complete the linear equation of $y = wx + b$. A bias is used for the ML algorithm to have another way to optimize the linear functions aside from the weights. The linear functions will also be inputted through an activation function as well and will take place right after the calculation of the linear function. III-A1 is a visual representation of a neural network.

The basic concept of a neural network can be explained by looking at a single neuron. First, the variables need to be defined as seen in Fig. 2. $a_n^{(k)}$ where n represents the neuron index and k represents the layer index. w will illustrate the weight. $w_{n,m}^{(k)}$, where n represents the neuron index from the previous layer and m represents the neuron index of the current layer. b_n^k will express the bias. z_n^k will signify the first part of

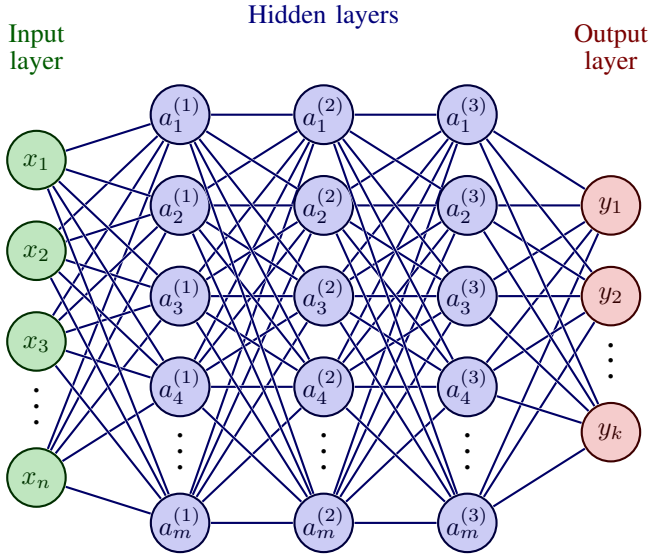


Fig. 1. A Deep Neural Network

the neuron, $\alpha(x) \in \mathbb{R}^1 \rightarrow \mathbb{R}^1$ (it will receive a scalar and output a scalar) will represent the activation function. z and α make up the entire neuron. As shown in Fig. 2 the preceding layer $a^{(1)}$ will enter the current layer's neuron where it will be multiplied by its respective w . The bias, through addition, enters the current layer's neuron. This will create the linear equation:

$$z_1^{(2)} = \vec{w} \cdot \vec{a}^{(1)} + b \quad (1)$$

Currently, this single neuron will only be able to make linear predictions however this project will demand a nonlinear predictor. To transform the linear equation to a nonlinear equation an activation function is used. The activation functions used within this project are the sigmoid function and ReLU function. The product z continues towards the activation function where non-linearity is introduced.

$$a_1^{(2)} = \text{ReLU}(z_1^{(1)}) \quad (2)$$

2) *Activation Functions*: The two activation functions are described as the following:

The sigmoid function:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

The ReLU function:

$$\text{ReLU}(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (4)$$

B. Stochastic Gradient Descent

Gradient descent is an optimization algorithm and as its name implies, gradient descent is an iterative process that revolves around calculating the gradient and using it to descend into a local minima. The process begins with the

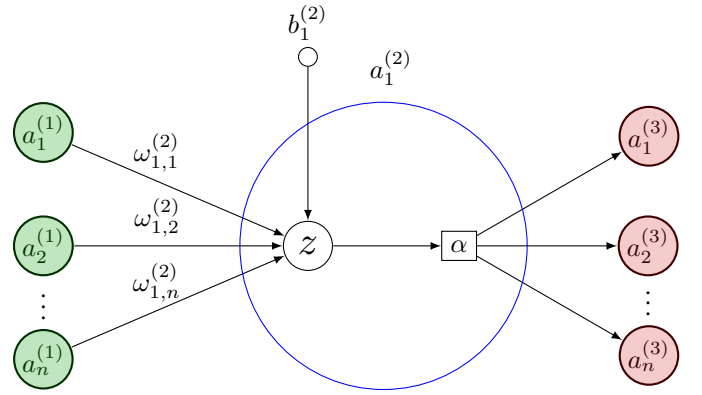


Fig. 2. A zoom in on a singular neuron

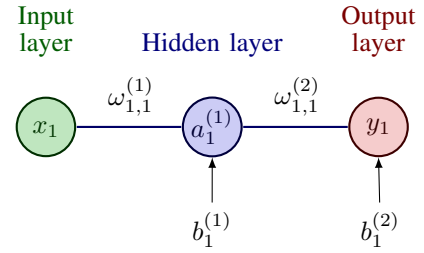


Fig. 3. Single neural network

calculation of the partial derivatives of the loss function with respect to the weights and biases. It will proceed with picking an initial value for its parameters and inputting them into the partial derivatives. The output will then be multiplied with the step size and used to update the weights and biases. Once they are updated the algorithm will iterate a specified amount of times known as *epochs*. into derivatives and keep updating them [6]. In this subsection the derivatives will be following a basic model for a neural network which is shown in Fig. 3

1) *Loss Function*: The *loss function* is a function composed of a hypothesis and the expected value. The loss function is a function which is inserted into the algorithm which is subsequently able to measure how accurate the prediction was. Different scenarios will demand different loss functions and in this project the loss function used is called Binary Cross Entropy.

2) *Binary Cross Entropy*: Binary cross entropy is a common loss function for classification-type problems. Classification type problems by convention have a loss function between the values "0" and "1" where each value will represent a different classification. Binary cross entropy achieves this with its usage of log functions. The binary part represents that the output will be either zero or one as opposed to a value ranging from zero to one. For this project the loss function will be structured as Eq. 5

$$L = \frac{1}{N} \sum_{i=1}^N -(y_i \cdot \log(p_i) + (1 - y_i) \cdot \log(1 - p_i)) \quad (5)$$

3) *The Partial Derivatives:* A vital step of stochastic gradient descent is to calculate the gradient of its current location so that it can move towards its local minima. Fig. 3 will be used as an example. It calculates the gradient through a process called back propagation.

Back propagation according to [7] is a fast converging algorithm that ANNs often implement because of its highly efficient reusing of already calculated partial derivatives in order to update the weights and biases. Back propagation uses the *chain rule* on $\frac{\partial L}{\partial w}$ and breaks it into several steps in the neural network.

$$\frac{\partial L}{\partial w_{1,1}^{(1)}} = \frac{\partial z_1^{(1)}}{\partial w} \frac{\partial a_1^{(1)}}{\partial z_1^{(1)}} \frac{\partial z_1^{(2)}}{\partial a_1^{(1)}} \frac{\partial y_1}{\partial z_1^{(2)}} \frac{\partial L}{\partial y_1} \quad (6)$$

Using the single neural network from Fig. 3, the equations used in order to solve them are as following:

$$z_1^{(1)} = w_{1,1}^{(1)} \cdot x_1 + b_1^{(1)} \quad (7)$$

$$a_1^{(1)} = \sigma(z_1^{(1)}) \quad (8)$$

$$z_1^{(2)} = w_{1,1}^{(2)} \cdot a_1^{(1)} + b_1^{(2)} \quad (9)$$

$$y_1 = \sigma(z_1^{(2)}) \quad (10)$$

$$L(y_1, y) = -(y_i \cdot \log(y_1) + (1 - y_i) \cdot \log(1 - y_1)) \quad (11)$$

The derivatives of these equations would then be calculated as such

$$\frac{\partial L}{\partial y_1} = -\frac{y}{y_1} + \frac{1 - y}{1 - y_1} \quad (12)$$

$$\frac{\partial y_1}{\partial a_1^{(2)}} = \sigma'(z_1^{(2)}) \quad (13)$$

$$\frac{\partial z_1^{(2)}}{\partial a_1^{(2)}} = w_{1,1}^{(2)} \quad (14)$$

$$\frac{\partial a_1^{(1)}}{\partial z_1^{(1)}} = \sigma'(z_1^{(1)}) \quad (15)$$

$$\frac{\partial z_1^{(1)}}{\partial a_1^{(1)}} = w_{1,1}^{(1)} \quad (16)$$

Once the derivatives have been calculated they will be used to update the weights and biases as shown in equations 19 and 20. This would conclude a single iteration and this will continue until a specified amount of iterations has been accomplished.

$$w_{1,1}^{(1)} = w_{1,1}^{(1)} - \alpha \cdot \frac{\partial L}{\partial w_{1,1}^{(1)}} \quad (17)$$

$$b_1^{(1)} = b_1^{(1)} - \alpha \cdot \frac{\partial L}{\partial b_1^{(1)}} \quad (18)$$

The $w_{1,1}^{(2)}$ and $b_1^{(2)}$ will also be updated in a similar fashion using the same process.

$$w_{1,1}^{(2)} = w_{1,1}^{(2)} - \alpha \cdot \frac{\partial L}{\partial w_{1,1}^{(2)}} \quad (19)$$

$$b_1^{(2)} = b_1^{(2)} - \alpha \cdot \frac{\partial L}{\partial b_1^{(2)}} \quad (20)$$

C. Vector summarization

To summarize shortly the notation for a deep neural network:

$$\begin{pmatrix} a_1^{(2)} \\ a_2^{(2)} \\ \vdots \\ a_m^{(2)} \end{pmatrix} = \sigma \left[\begin{pmatrix} w_{1,0} & w_{1,1} & \dots & w_{1,n} \\ w_{2,0} & w_{2,1} & \dots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m,0} & w_{m,1} & \dots & w_{m,n} \end{pmatrix} \begin{pmatrix} a_1^{(1)} \\ a_2^{(1)} \\ \vdots \\ a_n^{(1)} \end{pmatrix} + \begin{pmatrix} b_1^{(2)} \\ b_2^{(2)} \\ \vdots \\ b_m^{(2)} \end{pmatrix} \right]$$

$$a^{(2)} = \sigma \left(\mathbf{W}^{(1)} a^{(1)} + \mathbf{b}^{(2)} \right)$$

Where a indicates the neurons inside the hidden layer, the superscript represents the which column of the hidden layer of the neural network and the subscript representing the row of the column. w has two subscripts where the first one represents the column and the second one represents the row. Finally, b represents the bias and follows the superscript and subscript formation that a uses. σ represents the activation function of layer 2.

D. Binary Classification

Classification ANNs are a specific type of ANN which is often used in scenarios where one divides different responses from the neural network into different classes. Classification therefore outputs a single value indicating which class it belongs to. Binary classification, as the name suggests has two classes.

Binary classification evaluates the prediction made by the hypothesis and is categorized into one of four output types; a true positive, a true negative, a false positive, and a false negative. In this report, the term *accuracy* will be referred to, and points to the accuracy using the formula from Eq. 21.

TP = True Positive
TF = True Negative
FP = False Positive
FN = False Negative

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (21)$$

A true prediction means that prediction made was correct while a false label meant that the prediction was incorrect. Positive and negative labels indicates what the hypothesis' answer created. Take the example of the hypothesis returning a false positive from the data used in this project, this would mean that the hypothesis guessed that the water was contaminated but in reality it was not.

E. Pre-Processing

A neural network's performance and speed can be improved without changing anything within it, this is done by pre-processing of the input data. There are several methods and ideas to enhance the data-set.

1) *Least Square Estimation*: The least square estimation takes the sum of all the squared differences between the value generated from a model and expected value. In this report we use this function to create the data-set by finding the squared difference between the temperature and current response data. We do this by using the following:

$$\mathbf{b} = \mathbf{r} - \mathbf{A}\hat{\mathbf{x}}, \quad \mathbf{A}^T \mathbf{A} \hat{\mathbf{x}} = \mathbf{A}^T \mathbf{b} \quad (22)$$

Finally the data will be the residual that is calculated by

$$\mathbf{r} = \mathbf{b} - \mathbf{A}\hat{\mathbf{x}} \quad (23)$$

2) *Feature scaling*: The idea behind feature scaling comes from cases where the input data ranges massively. The problem is that features that are of much greater magnitude may be weighted or have more importance in the algorithm than what they really should. Feature scaling scales everything to the same range and as a consequence the algorithm uses each feature fairly. Another benefit to using feature scaling is that the flow of gradient descent is more fluid and more effective at reaching the minima. Feature scaling is done through a variety of normalization techniques and in this project linear normalization is used.

IV. METHOD

In this section each part of the implementation will be thoroughly explained in order to describe the process taken. Sub-section IV-A will explain how the data is constructed. Section IV-B will discuss the pre-processing of the data into a data-set and IV-C will discuss how the data was assembled before entering the algorithm.

This document will refer to the construction of each layer in the neural network through percentages. Where in a matrix $\mathbf{A}^{m \times n}$, where m corresponds to the amount of features and n corresponds to the amount of data points, each consecutive value corresponds a percentage of m . E.g., a matrix \mathbf{A} with the dimensions 600×100 and labeled as '70—50—1' will refer to a consecutive layer layout containing 70%, 50% and 1% of the amount of features respectively; which in this case corresponds to 70—50—1 neurons per layer.

A. Sensor data

Two data-sets were provided by Linköpings University. The first data-set comes from the Linghem pressure-boosting station, connected to the greater drinking water network of Linköping. One sample of data was recorded every minute over the span of 90 days. The second data-set comes from the

Nykvärn testing site, which is set in a controlled environment with small and slow variations. On this site, the controlled injection of sewage water into the pipe can take place. The data was recorded over a 4-day continuous span, with each measurement series recorded in a span of 10 seconds.

In Linghem, sewage water can not be injected as the station provides tap water. The data from Linghem is labeled as non-contaminated and consists of natural variations. The data consists of several measured features including the chlorine concentration, the inflow, and outflow of water, etc, at this site. At the Nykvärn testing site, temperature is the only environmental factor measured.

The *electric tongue* consists of three different electrodes made up of elements Au(Gold), Pb(Lead), and Rh(Rhodium), each taking their own measurements [8]. An electrode is an electrical conductor that makes contact with the nonmetallic circuit parts of a circuit, in this case water. The type of electrode used in the measurements depends on the contents of the aqueous solution being measured. An electrode has to be a good electrical conductor so it is usually a metal. The metal used in the electrode depends on the certain chemical properties of the metal, which in turn will result in different measurements as particular chemical reactions for their corresponding electrode takes place [9]. A voltage pulse train is applied and the current response to this voltage pulse train is measured. Each measurement series takes 10 seconds and in that time span the measurement creates a series of measurements taking 4 seconds and containing 4000 data points where each measurement takes 1ms totalling in $24\text{h} \times 60\text{min} \times 10\text{s} \times 4000 \text{ measurements} = 57600000$ measurements in total. Each of the 4000 data points received from the current response can be seen as to correspond to a single feature however each 'feature' can lead to responses that do not necessarily have the same behaviour. This meant that the inclusion of different features then increases the complexity of the problem [10].

In order to simplify the model, the point that gave the highest amperage of the electrode series was used from the 'Au' probe, being point 101 in Fig. 4. This was done as it was assumed that it would give us an accurate representation on the response of contamination as well decrease the computational stress on the network.

B. Pre-processing

Based on the research done by Skogsberg and Gelin [11] it was concluded that there was apart from the temperature very little correlation between the data coming from the features from Linghem and the current response. Temperature was shown to have had the most significant impact in Linghem. Therefore, temperature was chosen as the only feature. A concern surrounding the Linghem data-set was its measurements are of supposedly non-contaminated water as by its nature it shouldn't be contaminated. A lack of a clear contamination would mean that reliable identification would

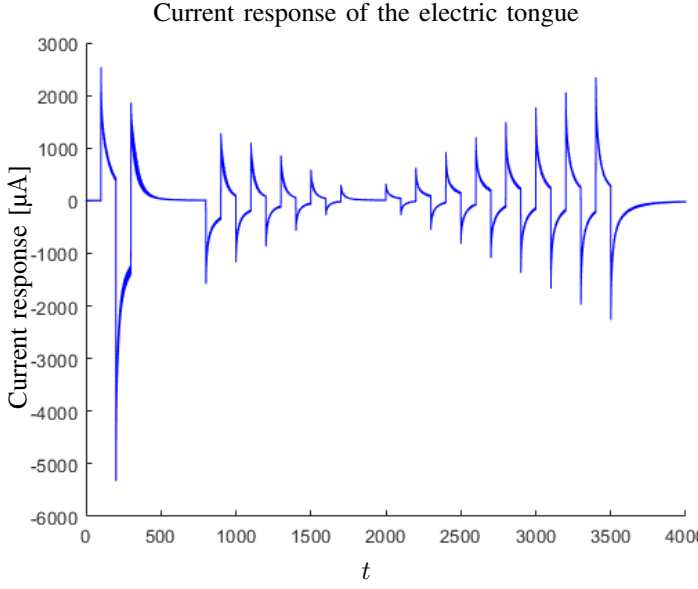


Fig. 4. Current response caused by a voltage pulse train in the electronic tongue

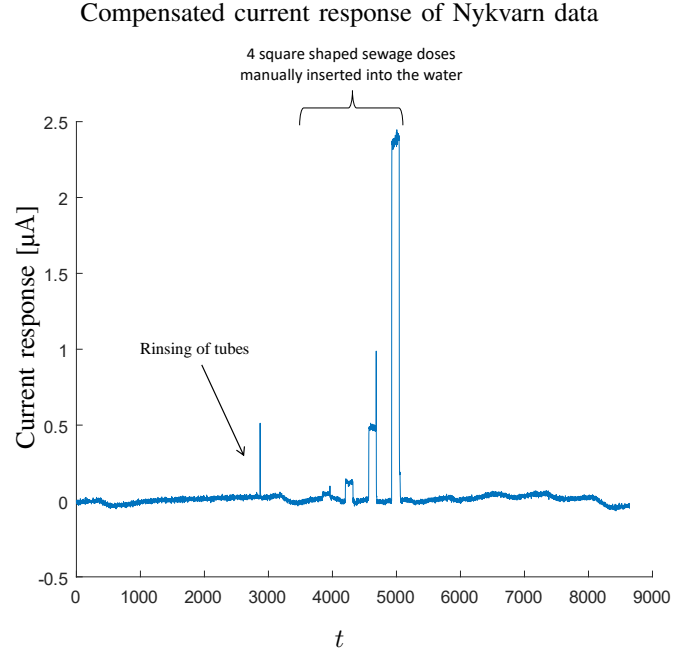


Fig. 6. Compensated measurement from Nykvarn at 2017-02-02.

be a significantly more difficult objective due to a lack of known contamination points.

Current response at Nykvarn with corresponding temperature

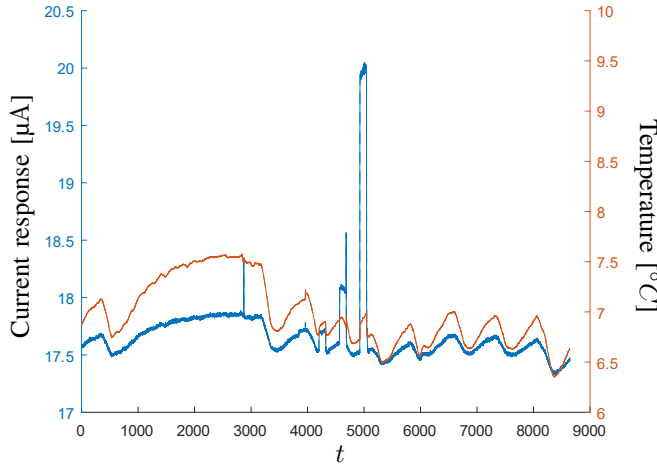


Fig. 5. Current measurement in blue; temperature measurement in orange. Measurements taken from Nykvarn at 2017-02-02. Note at time 3900, 4300, 4600 and 5000 have sewage water manually inserted with these percentages 0.05%, 0.23%, 0.67% and 3.4% respectively

As shown in diagram Fig. 5 there is a clear correlation between the temperature and the current response. For this reason it was decided upon to use the data from Nykvarn to build a model that would be able to detect peaks and deviations from what may be considered non-contaminated.

Temperature was compensated for using least square estimation as described in Section III-E1.

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & \dots & 1 & 1 \\ T_1 & T_2 & \dots & T_{n-1} & x_n \end{bmatrix}^T, \quad (24)$$

$$\mathbf{b} = [x_1 \ x_2 \ \dots \ x_{n-1} \ x_n]^T \quad (25)$$

Where T_s corresponds to the temperature at sample 's' and x_s corresponds to the data at sample 's'. From these matrices an approximation was calculated with

$$\mathbf{b} = \mathbf{r} - \mathbf{A}\hat{\mathbf{x}}, \quad \mathbf{A}^T\mathbf{A}\hat{\mathbf{x}} = \mathbf{A}^T\mathbf{b} \quad (26)$$

To retrieve the compensated data the residual was finally calculated with

$$\mathbf{r} = \mathbf{b} - \mathbf{A}\hat{\mathbf{x}} \quad (27)$$

The compensated data can be seen in Fig. 6 which reveals a strong correlation between temperature and the current response. This was the basis on the use of temperature as the only feature in the ANN.

C. Data assembly

The electric tongue's data-set can be visualized as the following matrix:

$$\mathbf{W} = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,n-1} & x_{1,n} \\ x_{2,1} & x_{2,2} & \dots & x_{2,n-1} & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{m-1,1} & x_{m-1,2} & \dots & x_{m-1,n-1} & x_{m-1,n} \\ x_{m,1} & x_{m,2} & \dots & x_{m,n-1} & x_{m,n} \end{bmatrix} \quad (28)$$

Where m denotes the the height of the matrix where each row corresponds to a measurement sample taken, this is taken over the course of one day. With a sampling period of 10 seconds this would ideally mean that there were $m = 24\text{h} \times 60\text{min} \times 6 + 1 = 8641$ samples per day however, due to a bug, there are inconsistencies on the amount of samples per day ranging from 8635 to 8641. For this reason measurements 8636–8641 were not included, leading to $m = 24\text{h} \times 60\text{min} \times 6 + 1 - 6 = 8635$ points from the electric tongue which were used. n denotes the length of the matrix where each column corresponds to one of the 4000 current response measurements taken in the span of 4 seconds. Notice that each day has its own vector meaning that there are 4 \mathbf{W} vectors in total. The data series from 28 was then created as

$$\mathbf{v}_{M,t} = \begin{bmatrix} (x_{M,1}) & (x_{M,2}) & \dots & (x_{M,t-1}) & (x_{M,t}) \end{bmatrix}^T \quad (29)$$

where M denotes which measurement point was used, as noted earlier point $M = 101$ was used, and t denotes the 10-second time interval where the measurement was taken.

In order to diversify the input data, two different input vectors were created; the first input vector had no compensation in regards to the temperature on the provided data points and another with where the data points had been compensated from temperature influence as described in Section IV-B. Both vectors have the same structure as Eq. 29.

The data-set was divided into two parts; a training data-set and a testing data-set. As dates, 2017-02-02 and 2017-02-04 contained examples of contamination peaks, the training data-set consisted of days 2017-02-02 and 2017-02-04; and the testing data-set consisted of days 2017-02-03 and 2017-02-05. This was done so as to have positive indications of contamination in both the training and testing data.

Linear normalization was used to normalize the vector $\mathbf{v}_{M,t}$. Linear normalization uses the following formula

$$\mathbf{A}_{i,norm} = \frac{\mathbf{v}_i - \min(\mathbf{v})}{\max(\mathbf{v}) - \min(\mathbf{v})} \quad (30)$$

The next step was to create an output vector. This vector had to represent contamination as well as when there was not a contamination. It was decided upon to use "1" to indicate a given sample corresponded to contaminated water and set all remaining samples to "0". This output matrix was then manually created.

$$\mathbf{v}_t = [\tau_t \quad \tau_t \quad \dots \quad \tau_t] \quad (31)$$

1) *Time window*: An ANN works best with a very large pool of data. Nykvarn, which consists of a 4-day period with 10 second sampling periods, became the sole source of data and due to the small amount of data it became necessary to extract as much data as possible from the data-set. Two methods were considered to expand the vectors to be used in the ANN such as using several more points from the current

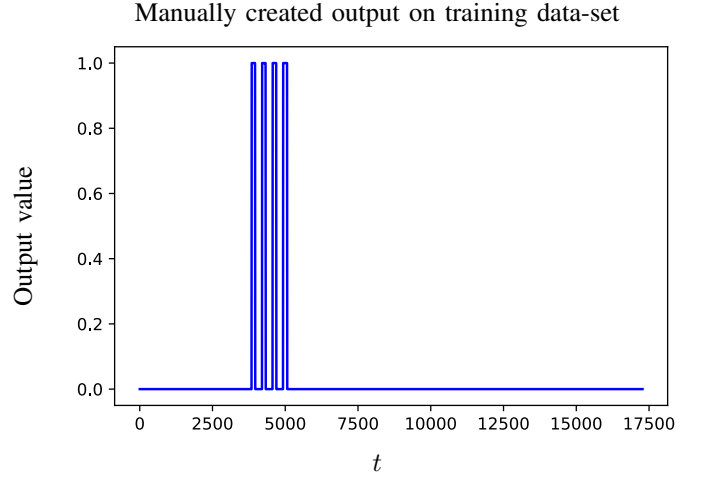


Fig. 7. Manually created output as a function of time. Note that all peaks indicate a contamination in the sytem. Produced from training data-set which include days 2017-02-02 and 2017-02-04

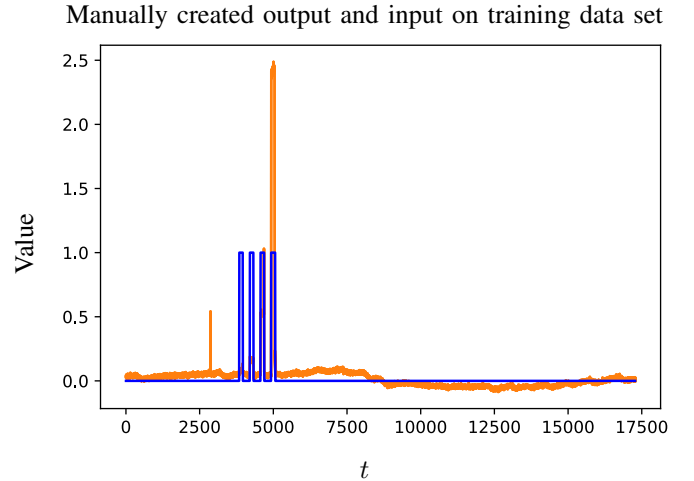


Fig. 8. Same output as Fig. 7 in blue; input for training data in orange

response as seen in Fig. 4 or creating a time window. The latter was concluded as a promising path forward as it would also hopefully remove the influence of the first peak as seen in Fig. 6 which isn't an example of contamination. This was because this first peak did not have the same breadth as the contaminated peaks as seen in Fig. 6 and therefore would lose its impact, the larger the time window became. This also created more features for the input of the ANN. Expanding the vectors using several more current responses was not used as that would have required knowledge on how each feature affects different measurements by the *electric tongue*.

It can also be noted from Fig. 6 that the injection of contamination into the system causes very significant and distinct square shaped peaks. Although not accurate in a realistic case, the assumption was necessary in the problem and would not have been necessary if more high-quality data was available. The shape and peaks were therefore, assumed to be an accurate representation of contamination in the

system.

In order for the ANN to learn temporal behaviours, a time window was created. This time window would gather all data from a time span. This would increase the width of the matrix which would hopefully create a more robust data-set. This created a vector $\nu_{M,t}$.

Table I reveals a couple of examples on the different structures of the time window. 'Before' refers to the measurements predating the current time in a certain interval while 'After' refers to the measurements subsequent the current time in a certain interval as visualized in Fig. 9.

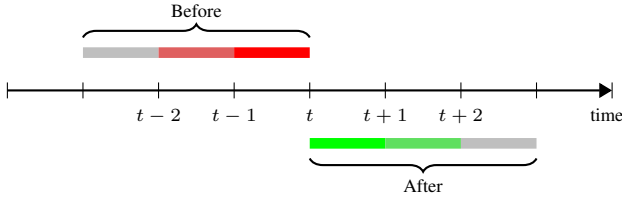


Fig. 9. A time window which includes 3 measurements before and 3 after

The results from different time windows can be seen in Table I.

TABLE I
LOSS AND ACCURACY FOR THE STRUCTURE '50—30—15—5'

Structure	Loss	Accuracy
'Before -0; After -0'	0.0265	0.9921
'Before -100; After -0'	0.0127	0.9970
'Before -0; After -100'	0.0162	0.9965
'Before -200; After -200'	0.0106	0.998
'Before -300; After -300'	0.0094	0.999

Loss functions value and accuracy for the corresponding time window structure. Note that the structure chosen here is an arbitrary one and not necessarily the final structure (although in this case it is).

D. Design of the classifier

To assemble the classifier, the Tensorflow [12] machine learning library and their neural network API Keras was used.

The classifier was created within the same design frame. However, as there are only two possible outcomes, the problem's output was reduced to two outputs; a 0 or 1, indicating respectively whether the water is non-contaminated or if it is. This is also known as a binary classification as mentioned in Section III-D.

V. RESULTS AND SIMULATIONS

The accuracy and loss from a lack of a time window can be seen in Fig. 10 and Fig. 11. As can be noted, the convergence rate is very high however the accuracy is relatively subpar and can be improved upon. As can be seen in Table I, the most successful result from testing different time windows became

Accuracy of ANN without a time window

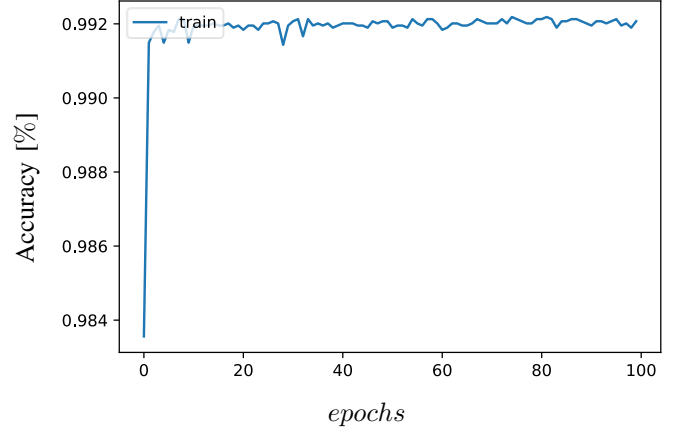


Fig. 10. 'Before -0; After -0'. The data is compensated. Notice that the convergence rate is very high, this can be attributed to the lack of time window present

Loss of compensated ANN without a time window

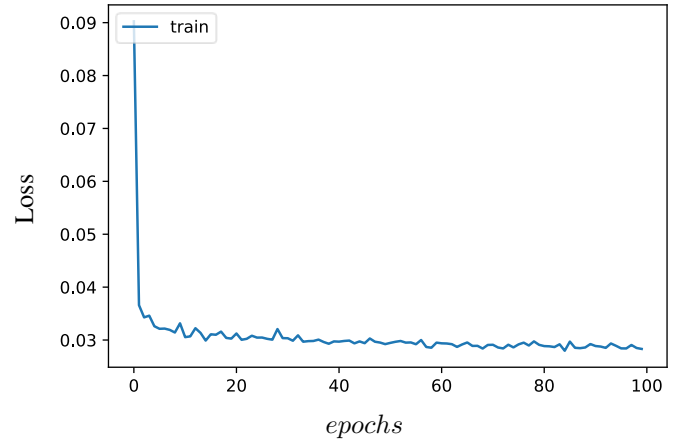


Fig. 11. 'Before -0; After -0'. As with Fig. 10 the data is compensated and the convergence rate is also attributable to the lack of a time window

a window of an equal spread of 600 samples. However, due to computational stress, a very similar result to an equal spread of a 400 total sample time window, as well as due to a risk for over-fitting, the latter was chosen for the classifier.

Both the uncompensated and the compensated data were tested upon using the 400 data points, evenly spread, time window. In order to try to achieve different results, several different structures of the hidden layers were created and tested on. The structures respective accuracy after 100 epochs is shown in Table II.

All layers used the *ReLU* activation function apart from the last output layer which used a *sigmoid* function.

From the results from Table II it is evident that the most promising results came from the compensated data as could be expected due to the nature of the graph as seen in Fig. 6. The

TABLE II
ACCURACY OF DIFFERENT DATA STRUCTURES

Structure (%)	Raw data	Compensated data
'50 – 30 – 15 – 5'	0.9983	0.9998
'80 – 60 – 40 – 20'	0.9978	0.9986
'90 – 65 – 40 – 15'	0.9919	0.9925
'75 – 50 – 50 – 30'	0.9934	0.9985
'90 – 50 – 25 – 25'	0.9962	0.9986

Loss functions value and the accuracy of a structure after 100 epochs of training with the time window structure 'Before –200; After –200'. Raw data are the raw measurements from the Nykvarn training data-set and compensated data suggests that the training data is pre-processed and compensated from the influence of temperature.

shape of which is very horizontal in nature, except for large protruding rectangular spikes. This is in contrast to the raw data, which as it was not compensated for temperature, had several more disturbances. The structure with the best accuracy was shown to be an ANN structure which consisted of its first hidden layer being of 50% of $m = 200$, and gradually decreasing to 30% followed by 15% with the final hidden layer being 5% of the original size. This was thus decided upon as the most promising structure.

A. Results

This project has lead to a supervised machine learning solution which is able to detect anomalies with great accuracy.

The input structure consists of temperature compensated data. Each point represents the value measured as a result of the peak current response which over a course of a day corresponds 8635 samples. Each vector is then adjusted for a time window creating 200 points per vector creating a vector which is assembled to a 8635×200 column vector.

The classifier consists of 4 hidden layers with the dimensional percentages of '50—30—15—5' in respect to the amount of features. This yields an accuracy and loss of $2 \times 10^{-3}\%$ as shown in Fig. 12 and 1.06×10^{-2} shown in Fig. 13.

On the tested data, the reconstruction produced by the algorithm is shown in Fig. 15. The first day in the testing data-set (where all the contaminations take place), which is compensated from temperature, is seen in Fig. 14. Notice there is a very small peak in Fig. 15, this is the remnants of the rinsing; a threshold could be implemented in order to remove such anomalies with relative ease.

VI. DISCUSSION

A. Unnatural Contamination Data

One of the concerns regarding the data-set was that it was created through a highly controlled environment, with contamination being added in an artificial and sudden manner resulting in the square shaped spikes that are seen in Fig. 6.

Accuracy of chosen ANN

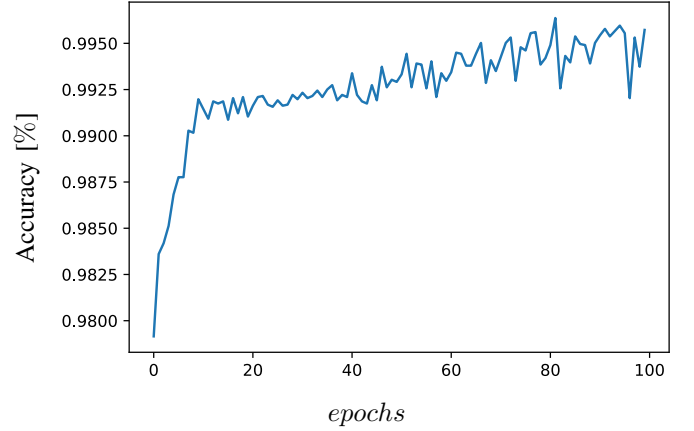


Fig. 12. Accuracy of chosen ANN ('50—30—15—5') on the test data as a function of number of epochs.

Loss of chosen ANN

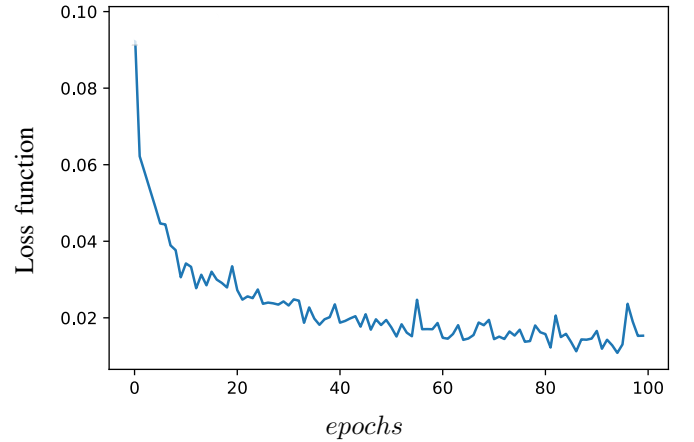


Fig. 13. Loss of chosen ANN ('50—30—15—5') on the test data as a function of number of epochs.

Whether the algorithm would be able to detect contamination in natural circumstances where drastic peaks indicating contamination are not necessarily a guarantee and where a slow rise and slow decrease is a possibility, would need to be tested further. The current algorithm looks at a time window and this is believed, with limitations, that given more data which can be labeled as well as adjusting said time window, the algorithm should be able to learn to detect these more natural shaped contamination, however this is yet to be tested.

B. Lingham data-set

The Lingham data-set contains various data-sets with different measurements; while most features did not reveal a close and immediate correlation, it still might be a data-set that can be useful if pre-processed in a different manner. One such potential method would be to pre-process the data from Lingham using a non-linear least square method instead of the linear least square method used in this study. Another technique would be to analyze different points in the current

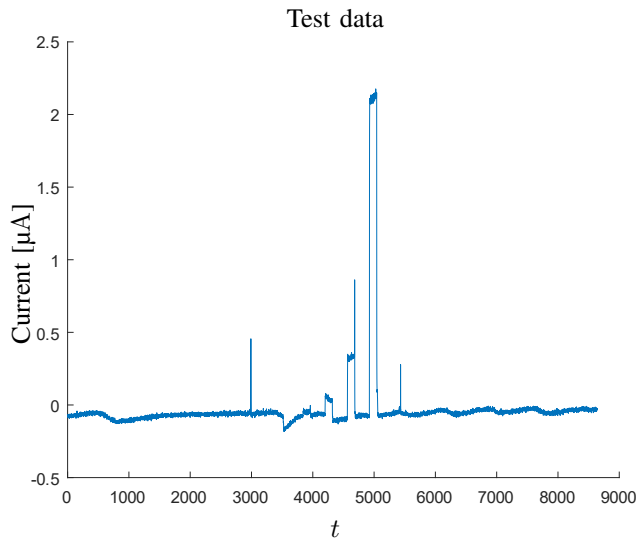


Fig. 14. Test data to be inserted into the neural network. Note, that the first peak and the last peak are not signs of contamination and are of rinsing; thus should not be regarded as such by the algorithm

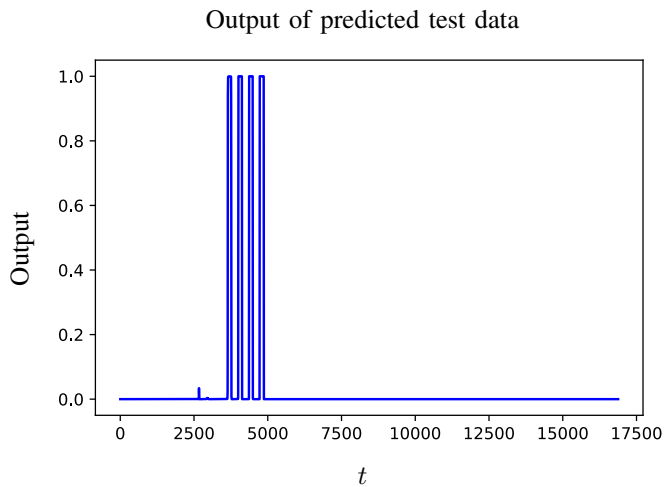


Fig. 15. Predicted output of test data with chosen ANN structure

response which may give different responses from certain features as Section VI-C mentions.

C. Additional Features

The feature used in this project was the highest point of the current response, refer to Fig. 4. In this case, the decision to use the highest current response infers that the data is highly dependent on the conductivity of the water, which has a high correlation with temperature. Temperature is an interesting feature to look at as contamination shows an identifiable pattern with it.

However, there are several more additional features that can be used as was done by Eriksson [10] where different parts of the signal were used as shown in Fig. 16. The reasoning behind the usage of this part of the signal was that it was less dependent on the conductivity and had a

strong dependency on redox activity. This information could be used, for example, but out of the scope of this problem, in the identification of different contamination.

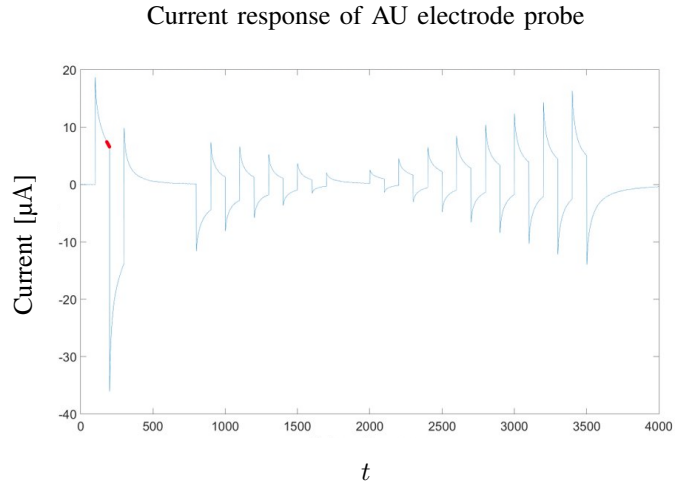


Fig. 16. The current response from [10] which points out a set of measurements (in red) that can be used to retrieve a different set of results from pre-processing

D. Evaluation of Accuracy

Two baselines on which the algorithm should beat were a random prediction and a linear algorithm. This surpasses both of them with great margin and a very high accuracy with the data from Nykvarn. The algorithm correctly removes the non-contaminating spike caused by the rinsing. However, as previously mentioned, this needs to be tested on a real data-set so as to avoid over-fitting.

E. Network sort

The largest weakness in using supervised learning is the attaining and the labeling of data. The manual processing of labeling data will require someone qualified to do it; however, it should only be needed in a controlled development phase where controlled experiments take place. However, this may also skew the results leading to different results in more controlled environments as opposed to more real-to-life environments. Training the algorithm from natural data might improve the algorithm. However, as the systems in place are designed to stop the occurrence of contamination as frequently as possible, labeling the data with great accuracy will be a challenge which is not solved with supervised learning.

In regards to the time window technique, there is a possibility of potential contamination being realized, depending on how large the window is. For example, if the window covers 100 data points, where 50 data points look into the past, 50 data points look into the future, and 1 data point signifies 10 seconds, it would take 500 seconds before the program will have discovered it. A solution to this problem would be to use a time window which only uses

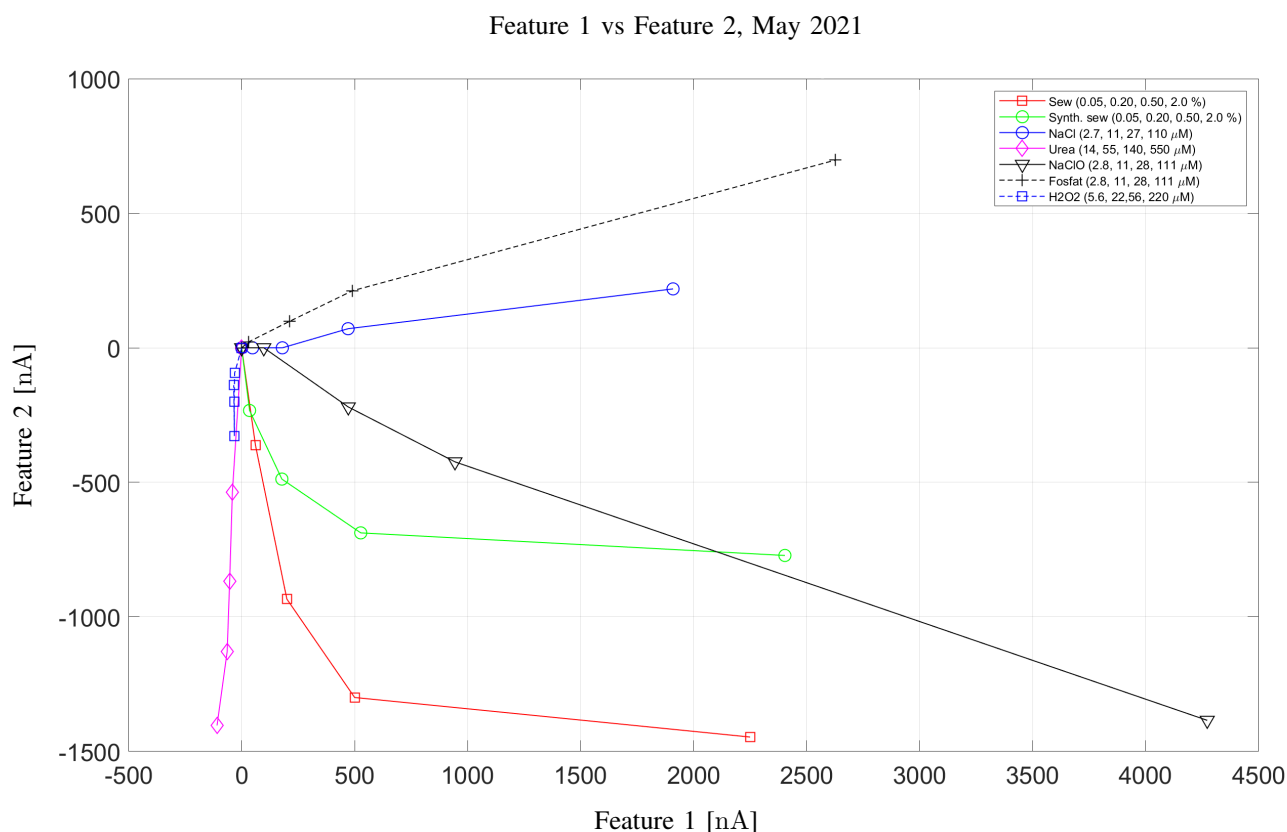


Fig. 17. Feature 1, used in this project, when compared to Feature 2 seen in Fig. 16 carried out in May 2021 by Mats Eriksson [10]

previous data points; however this was shown to be a weaker system in this report.

The time window used in this project was an equal distribution between before and after the specific measurement of 400 measurements. Despite an equal distribution of 600 yielding a better result, this was not chosen as it did not significantly improve the result, as well as an increased risk of over-fitting seemed to be a real possibility.

The binary classifier structure '50—30—15—5' was used. This was chosen as it had the highest accuracy as well as the smallest loss. Other structures were also viable candidates as loss and accuracy did not change drastically as a result of these changes. As increasing the complexity of the neural network did not seem to correspond to a greater accuracy or a smaller loss, the chosen classifier, which has fewer neurons than the other structures that were tested upon, seems like a viable structure. This does not necessarily mean that in a real life situation the chosen structure '50—30—15—5' would have been as useful in detecting anomalies. Larger or smaller neural network structures could have been more effective depending on the sensitivity of the data. If the neural network decreases in size, this could mean that the accuracy for the tested data decreases, however, may be more sensitive to anomalies in the data-set.

F. Further Research

This project has shown the potential of using supervised learning in detecting whether there has been water contamination. There are several different improvements that can be made upon this current study through further research on the identification of contamination and implementation changes.

1) Identifying the Contamination: One of the possible ways to further this research is to be able to identify what type of contamination the water is experiencing. Mats Eriksson presented the potential within using the entire data-set and looking at the complete signals. Using different parts of the signal resulted in clear patterns from different contaminants. This suggests that depending on how the signal is pre-processed, the algorithm may be improved and expanded upon.

The combination of these two features where feature 1 was plotted against feature 2 (again feature refers to, in this case, another point in Fig. 4) was a graph that presented various different compounds and elements as lines with their own unique pattern. There are strong suggestions that supervised learning could be used effectively in order to identify certain contaminants based on these patterns shown in Fig. 17.

2) *Implementation changes*: There are several solutions in improving the computational performance of the network. The largest impact will come from using the GPU, which can be done using Tensorflow. This would heavily improve the computational performance [13].

VII. CONCLUSION

This project goal was to examine whether binary classification was a viable ML method as a reliable tool for monitoring water quality. This was partially confirmed with the construction of a binary classifier using an ANN. This was tested through the usage of data, specifically Nykvarn, provided by Linköpings Universitet, from a sensor known as an 'electric tongue'. The data was first compensated for temperature using a linear least square estimation. This data was expanded upon to include measurements from a certain period before and after in the form of a 'time window' see Section IV-C1. Training was implemented on two different days and achieved an accuracy of 99.98% on the test data on two different days. There are several aspects that must be researched further, such as whether a classifier is a viable method in a more natural environment, testing different ANN constructions, and the usage of different points of the current response.

ACKNOWLEDGMENT

The authors would like to thank our Supervisor Henrik Hellström for his invaluable resources and tips. This project would not have gotten as far without his assistance. We would also like to thank Mats Eriksson from Linköpings Univseritet (LiU) for allowing access to all of his data which was most invaluable.

REFERENCES

- [1] Bagenstose, K. (2021, May) Dead snakes and mice, toxic sludge: How pathogens go unnoticed in america's water towers. Updated: June 2021. [Online]. Available: <https://eu.usatoday.com/in-depth/news/investigations/2021/05/21/infrastructure-neglect-water-towers-add-millions-illnesses/6769259002/>
- [2] L. Ramasamy and S. Kadry. Bristol, United Kingdom: IOP Publishing, May 2021.
- [3] C. Alfvén, H. Jonasson, D. Ilver, D. Lindgren, H. Winquist, F. and Stavk-lint, M. Asplund, M. Magounakis, M. Eriksson, N. Strömbeck, P. Jons-son, S. Mokhlesi, and T. Pettersson, *Elektronisk tunga och andra onli-nesensorer för detektion av föroreningar i dricksvattennätet*. Bromma, Stockholm: Svensktvatten, 2018.
- [4] Q. Liu and Y. Wu, "Supervised learning," 2012, doi: 10.1007/978-1-4419-1428-6_451.
- [5] M. Alloghani, D. Al-Jumeily Obe, J. Mustafina, A. Hussain, and A. Al-jaaf, *A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science*. Cham, Switzerland: Springer Publishing Company, Incorporated, Jan 2020, pp. 3–21.
- [6] J. Unpingco, *Python for Probability, Statistics, and Machine Learning*, 2nd ed. Cham, Switzerland: Springer Publishing Company, Incorporated, 2019.
- [7] H. Mohammed, A. Haithem, Alani, and L. George, "Comparison between resilient and standard back propagation algorithms efficiency in pattern recognition," *International Journal of Scientific and Engineering Research*, vol. 6, p. 773, 2015.
- [8] P. Ivarsson, C. Krantz-Rülcker, F. Winquist, and I. Lundström, "A Voltammetric Electronic Tongue," *Chemical Senses*, vol. 30, pp. 258–259, Jan 2005.
- [9] B. Losiewicz, G. Dercz, and M. Popczyk, "Electrode materials," *Solid State Phenomena*, vol. 228, pp. 3–15, Mar 2015.

- [10] M. Eriksson. (2022, Apr) Meeting-LiU-KTH-FOI apr 22, 2022. Univer-sity Lecture [PowerPoint slides]. Linköping: Linköpings Universitet.
- [11] R. F. Skogsberg and M. Gelin, "Water Contamination Detection With Artificial Neural Networks," Bachelor's Thesis, KTH, Stockholm, 2019.
- [12] F. Chollet *et al.* (2015) Keras. GitHub. [Online]. Available: [url{https://github.com/fchollet/keras}](https://github.com/fchollet/keras)
- [13] run.ai. (2021) Keras gpu. [Online]. Available: <https://www.run.ai/guides/gpu-deep-learning/keras-gpu#:~:text=With%20Run%3AAI-,Using%20Keras%20on%20a%20Single%20GPU,runs%20on%20GPU%20by%20default.>

Online Sample Selection for Resource-Constrained Networked Systems

Samuel Miksits and Philip Sjösvärd

Abstract—As more devices with different service requirements become connected to networked systems, such as Internet of Things (IoT) devices, maintaining quality of service becomes increasingly difficult. Large data sets can be obtained ahead of time in networks to train prediction models offline, however, resulting in high computational costs. Online learning is an alternative approach where a smaller cache of fixed size is maintained for training using sample selection algorithms, allowing for lower computational costs and real-time model re-computation.

This project has resulted in two newly designed sample selection algorithms, Binned Relevance and Redundancy Sample Selection (BRR-SS) and Autoregressive First, In First Out-buffer (AR-FIFO). The algorithms are evaluated on data traces retrieved from a Key Value store and a Video on Demand service. Prediction accuracy of the resulting model while using the sample selection algorithms and the time to process a received sample is evaluated and compared to the pre-existing Reservoir Sampling (RS) and Relevance and Redundancy Sample Selection (RR-SS) with and without model re-computation.

The results show that, while RS maintains the lowest computational overhead, BRR-SS outperforms both RS and RR-SS in prediction accuracy on the investigated traces. AR-FIFO, with its low computational cost, outperforms offline learning for larger cache sizes on the Key Value data set but shows inconsistencies on the Video on Demand trace. Model re-computation results in reduced error rates and significantly lowered variance on the investigated data traces, where periodic model re-computation overall outperforms change detection in practicality, prediction accuracy, and computational overhead.

Sammanfattning—Allteftersom fler enheter med olika servicekrav ansluts till nätverkssystem, såsom Internet of Things (IoT) enheter, ökar svårigheten att erhålla nödvändig servicekvalitet. Nätverk kan ge upphov till stora datamängder för träning av prediktionsmodeller offline, dock till en hög beräkningskostnad. Ett alternativt tillvägagångssätt är onlineinläring där en mindre cache av fast storlek upprätthålls för träning med hjälp av datapunkturvalsalgoritmer. Detta möjliggör lägre beräkningskostnader samt realtidsmodellräkningar.

Detta projekt har resulterat i två nydesignade datapunkturvalsalgoritmer, Binned Relevance and Redundancy Sample Selection (BRR-SS) och Autoregressive First In, First Out-buffer (AR-FIFO). Algoritmerna utvärderas på dataspar som hämtats från ett Key Value-lager och en Video on Demand-tjänst. Förutsäggelseförmåga för den resulterande modellen när datapunkturvalsalgoritmerna används och tid för bearbetning av mottagen datapunkt utvärderas och jämförs med dem redan existerande Reservoir Sampling (RS) och Relevance and Redundancy Sample Selection (RR-SS), med och utan modellräkning.

RS resulterar i lägst beräkningskostnad medan BRR-SS överträffar både RS och RR-SS i förutsäggelseförmåga på dem undersökta spåren. AR-FIFO, med sin låga beräkningskostnad, överträffar offlineinläring för större cachestorlekar på Key Value-spåret, men visar inkonsekvent beteende på Video on Demand-spåret. Modellräkning resulterar i mindre fel och avsevärt sänkt varians på dem undersökta spåren, där periodisk modellräkning totalt sett överträffar förändringsdetektering i praktikalitet, förutsäggelseförmåga och beräkningskostnad.

Index Terms—Online learning, Sample Selection, Real-time Learning, Random Forest, Reservoir Sampling, Relevance and Redundancy, Binned Distribution, Autoregressive Model, Change Detection, Model Re-computation.

Supervisors: Rolf Stadler and Xiaoxuan Wang

TRITA number: TRITA-EECS-EX-2022:164

I. INTRODUCTION

As society is becoming more and more interconnected, the need to uphold robust quality of service within networks will become greater than ever. Since different types of devices, such as “Internet of Things” (IoT) devices, will share network infrastructure with already existing devices and other further additions, the difference in required service could become substantial [1].

Monitoring a networked system to create a large enough data set is costly, not to mention the cache size required to store said data and the computational power to process it [2]. Therefore in a resource-constrained environment, constructing a large data set ahead of time to train a machine learning model, referred to as offline learning, is not always viable. Using sample selection algorithms to remove irrelevant and redundant monitored samples can allow for online learning, where smaller sets of data reduce the computational overhead by reducing model training time and storage requirements. Furthermore, data collection in real-time is also possible. Aside from reducing the computational overhead, the goal of online learning is also to achieve similar prediction accuracy for the machine learning model trained on the cache to that of offline learning.

This project aims to build upon the research done on sample selection algorithms found in [2] to explore new selection methods. Additionally, investigated in the project is also the impact of model re-computation for online learning, a process made possible by the small training set size. Investigated is both the use of periodic model re-computation and change detection to trigger the model re-computation.

II. BACKGROUND

There have been numerous works of literature exploring the topic of online feature selection algorithms [2]. The work of online sample selection algorithms is, however, limited. Literature exploring the use of few samples with higher dimensions is often not widely applicable to the goal of this project, that is, online sample selection algorithms for resource-constrained infrastructures. The literature found in

[2], which the project builds upon, explores several online sample selection algorithms and how they compare. Found among the different algorithms are both supervised and unsupervised sample selection methods. For unsupervised sample selection algorithms, during the processing of a received sample from the monitored system, considered is only the data stream \mathbf{X} . A supervised sample selection algorithm has additional access to the corresponding target values \mathbf{Y} during selection.

Investigating the use of model re-computation is important as it can leverage the flexibility of online learning to reduce the risk of the model not corresponding to the current state of the data trace. The idea of “concept drift” is thoroughly explored in [3] and is described as distribution varying over time in data streams. Concept drift is analogized in [3] to predicting the sales of clothes, and if clothes are more frequently sold during the summer, then using data from the winter months may cause the prediction accuracy to suffer. “Video on Demand” (VoD) services have, in a similar vein to the sales of clothes, differences in traffic during different hours of the day, and as such, model re-computation becomes important.

III. METHODOLOGY

A. Data Preprocessing

The goal of the project is online sample selection, where the aim is to reduce the sample set and thereby also the computational overhead. The investigated traces from [4] are denoted “KV flashcrowd - JNSM 2017” and “VoD periodic - CNSM 2015” using target values “ReadsAvg” and “DispFrames” respectively, see Figure 1 for visualization. These target values represent the average response time of read operations each second and the number of displayed video frames per second on the client side, respectively.

Since the investigated traces contain over 1000 features, reducing the feature set is an important preprocessing step for achieving efficient evaluations of the online sample selection algorithms. Therefore, a Random Forest Regressor from the Scikit-learn library [5] with 20 trees is used to find a sorted list of the 16 most important features for each respective trace [6], see Appendix A for used features. For all further evaluations, only these 16 features are considered. Furthermore, all values in \mathbf{X} are standardized to have a 0 mean and a variance of 1, where samples containing a feature with an absolute value over 3 are viewed as outliers and are removed [6].

B. Re-computation of Model

The first investigated model re-computation method is periodic model re-computation, where the model is re-computed every T_c received sample.

The second method is to use a change detection method such as “Student-Teacher Method for Unsupervised Concept Drift Detection” (STUDD), presented in [7]. STUDD uses a teacher and student model, where a data set \mathbf{X}_{tr} with target values \mathbf{Y}_{tr} is used to train the former. After training the teacher model, the test set of the model is the same data set used for training, where the resulting prediction of \mathbf{Y}_{tr} , is denoted $\hat{\mathbf{Y}}_T$. Trained using \mathbf{X}_{tr} is also the student model, however, using $\hat{\mathbf{Y}}_T$ as

target values instead of \mathbf{Y}_{tr} . Then the student model computes its student prediction denoted by $\hat{\mathbf{Y}}_S$.

Change detection using STUDD, as explained in [7], is handled by using the Page-Hinkley test with $\delta = 0.001$ [8], $\lambda = 50$, $\alpha = 0.9999$ and a minimum of 30 samples before a change is detected [9]. The input parameter for the Page-Hinkley test is, in this case, the error rate between each prediction in $\hat{\mathbf{Y}}_S$ and $\hat{\mathbf{Y}}_T$. If the Page-Hinkley test detects a change, then the latest 1000 samples are used to re-compute the student and teacher model.

C. Evaluating Prediction Accuracy of Online Learning Methods

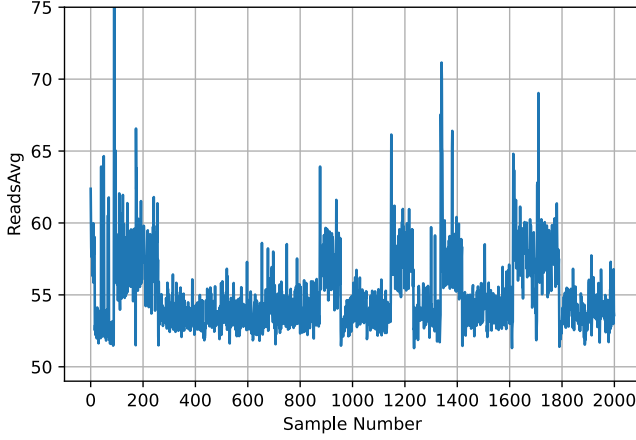
Evaluation of prediction accuracy for online learning algorithms on existing traces is performed through 20 independent runs where each run uses a uniformly random starting point t_0 on the trace and fills the cache of size N with samples $(\mathbf{X}_{t_0}, \mathbf{Y}_{t_0}), \dots, (\mathbf{X}_{t_0+T_p-1}, \mathbf{Y}_{t_0+T_p-1})$ according to the respective sample selection algorithm [6]. T_p denotes the number of initial samples to be processed to construct the initial cache. Used for all further evaluations of prediction accuracy is $T_p = 3000$. Furthermore, used across all algorithms is the same set of starting points.

After the initial cache has been created, a Random Forest Regressor using 20 trees is trained on the cache, where the test set used for evaluating the model is all subsequent samples on the trace, $(\mathbf{X}_{t_0+T_p}, \mathbf{Y}_{t_0+T_p}), (\mathbf{X}_{t_0+T_p+1}, \mathbf{Y}_{t_0+T_p+1}), \dots$ that is. If periodic model re-computation is used, the test set samples are instead $(\mathbf{X}_{t_0+T_p}, \mathbf{Y}_{t_0+T_p}), \dots, (\mathbf{X}_{t_0+T_p+T_c-1}, \mathbf{Y}_{t_0+T_p+T_c-1})$. The current cache is then updated by processing the T_c next samples, followed by a new model trained with the newly updated cache [8]. In other words, the newly processed samples are the same ones used for the previous test set. The process is then continuously repeated for the next T_c samples until the end of the trace. For clarity, the second test set of samples is $(\mathbf{X}_{t_0+T_p+T_c}, \mathbf{Y}_{t_0+T_p+T_c}), \dots, (\mathbf{Y}_{t_0+T_p+2T_c-1}, \mathbf{Y}_{t_0+T_p+2T_c-1})$.

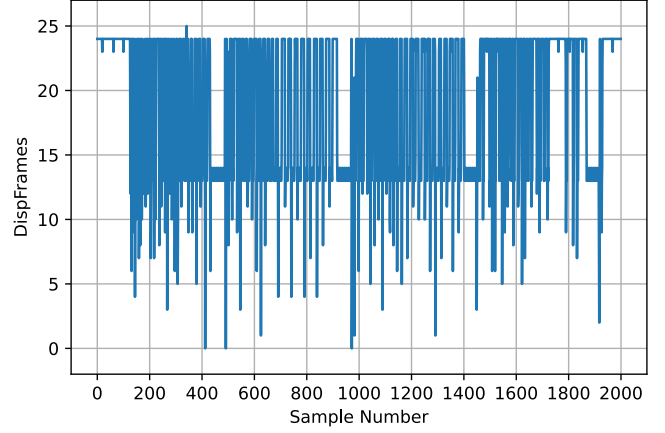
Evaluation of prediction accuracy using change detection is similar to that of periodic model re-computation. The initial model is trained after processing the first T_p samples starting from t_0 . Each newly received sample after $(\mathbf{X}_{t_0+T_p-1}, \mathbf{Y}_{t_0+T_p-1})$ is processed by the given sample selection algorithm, potentially updating the cache. If sample $(\mathbf{X}_t, \mathbf{Y}_t)$ causes a change to be detected, then the test set samples used for evaluating the initial cache are $(\mathbf{X}_{t_0+T_p}, \mathbf{Y}_{t_0+T_p}), (\mathbf{X}_{t_0+T_p+1}, \mathbf{Y}_{t_0+T_p+1}), \dots, (\mathbf{X}_{t-1}, \mathbf{Y}_{t-1})$ [8]. After evaluation, the teacher, student, and prediction model are re-computed, all being Random Forest Regressors using 20 trees. The process is then, similar to periodic model re-computation, repeated until the end of the trace.

Evaluating the prediction accuracy of the sample selection algorithms is performed by calculating the resulting Normalized Mean Absolute Error (NMAE) of each corresponding test set. The metric is defined as

$$NMAE = \frac{\sum_{t=1}^m |\hat{\mathbf{Y}}_t - \mathbf{Y}_t|}{\sum_{t=1}^m |\mathbf{Y}_t|}, \quad (1)$$



(a) "KV flashcrowd - JNSM 2017."



(b) "VoD periodic - CNSM 2015."

Fig. 1. Time series plot of the first 2000 samples of the investigated traces for each target value.

where \hat{Y}_t is the predicted value of the corresponding target value Y_t , and m is the size of the set [2]. The value of the NMAE is computed individually over the 20 different runs and then presented as a mean value with a 95% confidence interval.

IV. SAMPLE SELECTION ALGORITHMS

Presented in this section are the four algorithms used in this project. Used as a baseline for the two new algorithms resulting from this project are two previously existing sample selection algorithms. The two previous ones are Reservoir Sampling and Relevance and Redundancy Sample Selection. The two newly designed ones are Binned Relevance and Redundancy Sample Selection and Autoregressive First In, First Out-buffer.

A. Reservoir Sampling

Reservoir Sampling (RS) is an unsupervised sample selection algorithm that dynamically caches N samples from \mathbf{X} [2]. Seen in Algorithm 1 is a detailed process of RS where \mathbf{C} is the cache and \mathbf{C}_r is the sample of index r in \mathbf{C} . The cache is initialized by the first N samples, see Lines 5 and 6. After initialization, updates to the reservoir with new samples are performed based on an algorithm where the t :th sample \mathbf{X}_t from the data stream \mathbf{X} is selected with the probability $\frac{N}{t}$ [10]. The sample to be replaced is randomly chosen from the N stored samples, resulting in that all received samples have the same probability of being included in the final cache. In terms of cache size and feature count, the algorithm has a computational complexity of $\mathcal{O}(1)$ as changes to the cache size or feature count do not change the number of computations.

B. Relevance and Redundancy Sample Selection

Relevance and Redundancy Sample Selection (RR-SS) is an unsupervised algorithm that compares received samples to a relevance and a redundancy list to maximize the inclusion of relevant samples and minimize redundant ones. Shown in

Algorithm 1 Create cache of size N using Reservoir Sampling.

Input: $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_t, \dots\}$
Output: Cache \mathbf{C}

- 1: $\mathbf{C} \leftarrow$ Empty cache of size N
- 2: $t \leftarrow 1$
- 3: **loop**
- 4: $\mathbf{X}_t \leftarrow$ Newly received sample
- 5: **if** $t \leq N$ **then**
- 6: $\mathbf{C}_t \leftarrow \mathbf{X}_t$
- 7: **else**
- 8: $r \leftarrow$ Random integer $1, 2, \dots, t$
- 9: **if** $(r \leq N)$ **then**
- 10: $\mathbf{C}_r \leftarrow \mathbf{X}_t$
- 11: **end if**
- 12: **end if**
- 13: $t \leftarrow t + 1$
- 14: **end loop**

Algorithm 2 is a detailed overview of the algorithm. The cache can be seen as an $N \times k$ matrix \mathbf{C} with N samples and k features [2]. In the cache \mathbf{C} , a sample vector is denoted \mathbf{C}_i with a sample, or row, of index $i = 1, 2, \dots, N$. A feature vector, that is a vector containing one feature across all samples, is denoted $\mathbf{C}_{:,j}$, where $j = 1, 2, \dots, k$ is the index of the feature or column. Furthermore, $\overline{\mathbf{C}}_{features}$ defines a vector with the average of each feature, that is

$$\overline{\mathbf{C}}_{features} = \{\overline{\mathbf{C}}_{:,1}, \overline{\mathbf{C}}_{:,2}, \dots, \overline{\mathbf{C}}_{:,k}\}. \quad (2)$$

Whether or not a sample is selected depends on its rank, calculated using $\text{Rank}()$ on Line 19. The rank is given by the relation between the relevance and redundancy list calculated by the average of the Euclidean distance $\text{EuclideanDist}()$ and the average of the cosine similarity $\text{CosineSim}()$ between each pair of sample vectors respectively. The first step after initializing the original cache is to create the two rank values $\text{rankNew} = \text{Rank}(\mathbf{X}_t^T, \mathbf{C}^T)$ and $\text{rankAvgSample} =$

$\text{Rank}(\bar{\mathbf{C}}_{features}^T, \mathbf{C}^T)$. The second step is to determine if sample \mathbf{X}_t is to be added to the cache, where \mathbf{X}_t is selected if $rankNew$ is greater than $rankAvgSample$, see Line 11. If selected, a rank list $\text{cacheRankList} = \text{Rank}(\mathbf{C}^T, \mathbf{C}^T)$ with the rank of all samples in \mathbf{C} is created. The sample that is to be replaced is the sample with the lowest rank in cacheRankList . The process is then repeated as a new sample is received.

Since the algorithm has to perform the pairwise operations from each sample vector in the cache to all sample vectors in the cache to compute the cacheRankList , the algorithm scales quadratically with the cache size. Additionally, the operations performed contain scalar products, resulting in it also scaling with the feature count. The resulting complexity for the sample selection algorithm is $\mathcal{O}(kN^2)$, where N is the cache size and k is the number of features.

Algorithm 2 Create cache of size N using Relevance and Redundancy Sample Selection.

Input: $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_t, \dots\}$

Output: Cache \mathbf{C}

```

1:  $\mathbf{C} \leftarrow$  Empty cache of size  $N$ 
2:  $t \leftarrow 1$ 
3: loop
4:   if  $t \leq N$  then
5:      $\mathbf{C}_t \leftarrow \mathbf{X}_t$ 
6:   else
7:      $\bar{\mathbf{C}}_{features} \leftarrow \{\bar{\mathbf{C}}_{:,1}, \bar{\mathbf{C}}_{:,2}, \dots, \bar{\mathbf{C}}_{:,N}\}$ 
8:      $\mathbf{X}_t \leftarrow$  Newly received sample
9:      $rankNew \leftarrow \text{Rank}(\mathbf{X}_t^T, \mathbf{C}^T)$ 
10:     $rankAvgSample \leftarrow \text{Rank}(\bar{\mathbf{C}}_{features}^T, \mathbf{C}^T)$ 
11:    if  $rankNew > rankAvgSample$  then
12:       $\text{cacheRankList} \leftarrow \text{Rank}(\mathbf{C}^T, \mathbf{C}^T)$ 
13:       $i \leftarrow$  Index of sample with lowest rank
        in  $\text{cacheRankList}$ 
14:       $\mathbf{C}_i \leftarrow \mathbf{X}_t$ 
15:    end if
16:  end if
17:   $t \leftarrow t + 1$ 
18: end loop
19: Procedure  $\text{Rank}(\text{matrix} \in \mathbb{R}^{i \times j}, \text{cache} \in \mathbb{R}^{n \times k})$  :
20:    $\text{redundanceList} \leftarrow \text{CosineSim}(\text{matrix}, \text{cache})$ 
21:    $\text{relevanceList} \leftarrow \text{EuclideanDist}(\text{matrix}, \text{cache})$ 
22:    $\text{rankList} \leftarrow \frac{\text{relevanceList}}{\text{redundanceList}}$ 
23:   return  $\text{rankList}$ 
24: end
```

C. Binned Relevance and Redundancy Sample Selection

The first new alternative sample selection algorithm that results from this project is the unsupervised Binned Relevance and Redundancy Sample Selection (BRR-SS). The concept behind the algorithm is to leverage the strengths of RS and RR-SS while minimizing their negative aspects. RS has the capability of maintaining the distribution of samples found on the data trace due to its randomness during selection, meaning that the percentage of target values from samples in the cache

for a specific range of values is close to that of the entire trace. The negative aspect of RS is that it lacks sample selection based on quality since all of its samples are randomly selected.

RR-SS improves on the selection aspect by actively choosing samples that it finds relevant while reducing the number of redundant ones. The downside of this approach is that the distribution of samples within the cache can differentiate from that of the entire data trace, resulting in poor prediction accuracy. BRR-SS seeks to achieve the distribution of RS while having a selection process with aspects in common to that of RR-SS. BRR-SS uses bins to maintain the distribution by tracking the number of samples received for each range of values, allowing the distribution of the cache to be adjusted to resemble that of all samples received so far.

The most natural way to decide which bin a received sample belongs to is to use the target value \mathbf{Y}_t as this would result in optimal distribution. However, since BRR-SS builds on concepts from RR-SS best suited for unsupervised sample selection, the value used is instead the most important feature of \mathbf{X}_t , that is $\mathbf{X}_{t,1}$. From the 16 used features, the first feature is the most important one since they were sorted based on importance during the preprocessing. For all further evaluations of BRR-SS, the first feature is, unless otherwise specified, used. Since the range of possible values for $\mathbf{X}_{t,1}$, meaning the range which the bins have to accommodate, is not known, the size of the bins has to be dynamically changed to make sure that all received samples have a bin that represents their value.

Algorithm 3 shows the procedure of BRR-SS where $numBins$, the number of bins used, is a multiple of 4. The algorithm begins with initializing a bin size $binSize$ that is intentionally too small to fit all received samples, see Line 3 where 0.0001 is used for all further evaluations. Following is the creation of a list, binsTotal using index $1, 2, \dots, numBins$, effectively functioning as bins to keep track of the number of samples received for each range of values. Alongside binsTotal is another set of bins, binsSamples , that corresponds to the former but instead contains lists of references to the samples in the cache belonging to each respective bin in binsTotal .

Seen in Algorithm 4 is the calculation of the bin number, $binNum$, of the received sample where $target$ is the value used to determine which bin the sample belongs to, which is $\mathbf{X}_{t,1}$, the value of the first feature unless otherwise specified. If $binNum$ ends up being outside the range of bins in binsTotal , the bin size will be adjusted by doubling the value of $binSize$ and merging each pair of bins in binsTotal and binsSamples . The merging process can be seen in Algorithm 5 and is repeated until $binNum$ can fit within the boundaries of binsTotal . An illustration of the merging process can be seen in Figures 2 and 3. The merging of bins ensures that the bins still represent the correct ranges, with the difference being that the ranges double in size each time.

Each newly received sample is added to the cache, with a reference to itself being added to the corresponding bin in binsSamples and the corresponding bin in binsTotal being incremented by 1. The first N samples fill the original cache, and when it is full, the average of the value used to

Algorithm 3 Create cache of size N using BRR-SS.**Input:** $\{X_1, X_2, \dots, X_t, \dots\}, numBins$ **Output:** Cache C

```

1:  $C \leftarrow$  Empty cache of size  $N$ 
2:  $t \leftarrow 1$ 
3:  $binSize \leftarrow$  Small number greater than zero
4:  $binsTotal \leftarrow$  List of size  $numBins$  with zeros
5:  $binsSamples \leftarrow$  List of size  $numBins$  with empty lists
6: loop
7:    $X_t \leftarrow$  Newly received sample
8:   Add  $X_t$  to  $C$ 
9:    $binNum \leftarrow$  Algorithm 4 with  $target \leftarrow X_{t,1}$ 
10:  while  $binNum > numBins$  or  $binNum < 1$  do
11:     $binSize \leftarrow 2 \cdot binSize$ 
12:     $binsTotal \leftarrow$  Algorithm 5
      with  $binsToMerge \leftarrow binsTotal$ 
13:     $binsSamples \leftarrow$  Algorithm 5
      with  $binsToMerge \leftarrow binsSamples$ 
14:     $binNum \leftarrow$  Algorithm 4 with  $target \leftarrow X_{t,1}$ 
15:  end while
16:  Add reference to  $X_t$  to  $binsSamples[binNum]$ 
17:   $binsTotal[binNum] \leftarrow binsTotal[binNum] + 1$ 
18:  if  $t = N$  then
19:     $binsMid \leftarrow \bar{C}_{:,1}$ 
20:  else if  $t > N$  then
21:     $greatestDeviation \leftarrow 0$ 
22:    for  $i = 1$  to  $numBins$  do
23:       $lengthBin \leftarrow$  Length of  $binsSamples_i$ 
24:       $deviation \leftarrow \frac{lengthBin}{binsTotal_i}$ 
25:      if  $deviation > greatestDeviation$  then
26:         $greatestDeviation \leftarrow deviation$ 
27:         $binSel \leftarrow binsSamples_i$ 
28:      end if
29:    end for
30:     $redundanceList \leftarrow \text{CosineSim}(binSel, binSel)$ 
31:     $relevanceList \leftarrow \text{EuclideanDist}(binSel, binSel)$ 
32:     $binRankList \leftarrow \frac{relevanceList}{redundanceList}$ 
33:     $sampleToRemove \leftarrow$  Sample of lowest rank
      in  $binRankList$ 
34:    Remove reference to  $sampleToRemove$ 
      from  $binsSamples[binNum]$ 
35:    Remove  $sampleToRemove$  from  $C$ 
36:  end if
37:   $t \leftarrow t + 1$ 
38: end loop

```

decide the bin number will in the original cache be calculated, in this case, the first feature. The result is $binsMid$, which permanently denotes the value used to mark the middle of the range of bins, see Line 19 where $\bar{C}_{:,1}$ is a feature vector of feature 1 in cache C . Otherwise, the bins would center around 0, which would be inefficient if, for example, $X_{t,1} > 0$ for all t as at least half of the bins would go unused.

If the cache is overfull after adding a newly received sample X_t , that is if $t > N$ on Line 20, a sample will have to be removed. This is done by comparing the distribution of

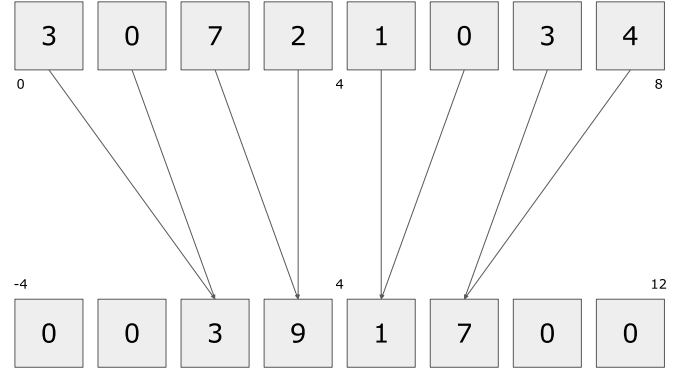


Fig. 2. Illustration of how $binsTotal$ is merged after 20 received samples using 8 bins where $binsMid = 4$ and $binSize$ changes from 1 to 2.

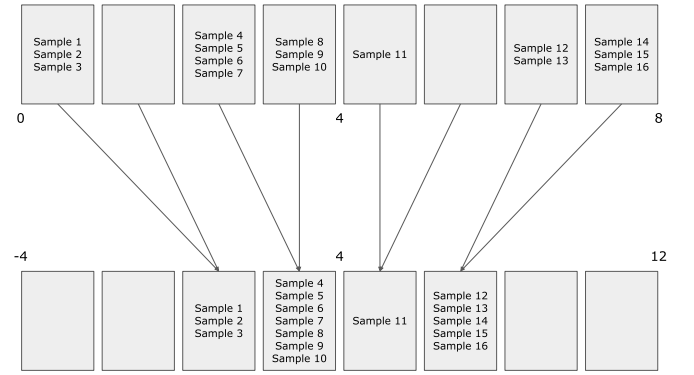


Fig. 3. Illustration of how $binsSamples$ is merged using 8 bins and a cache size of 16 where $binsMid = 4$ and $binSize$ changes from 1 to 2.

samples across each bin in $binsSamples$ to that of the distribution in $binsTotal$, see Lines 22-29. The bin selected to have one of its samples removed is $binSel$ which has the largest $deviation$ value, meaning the bin that has too many samples when compared to the total distribution of samples received so far.

Algorithm 4 Calculate bin for received sample in BRR-SS.**Input:** $target, binSize, binsMid, numBins$ **Output:** Calculate $binNum$

```

1:  $binNum \leftarrow \frac{target - binsMid}{binSize} + \frac{numBins}{2}$ 
2: return  $binNum$ 

```

The sample to be removed within $binSel$ is decided by using a similar approach to RR-SS, however differentiating on a fundamental level. BRR-SS does not compute $rankNew$ or $rankAvgSample$ found in Algorithm 2. Furthermore, when BRR-SS uses a rank list, it calculates the rank list of only the selected bin instead of the entire cache to reduce the computational overhead. The removed sample is the sample of the lowest rank, which could end up being a sample from the cache before X_t was added or X_t itself. Since BRR-SS uses the rank list from RR-SS, the complexity for both of the algorithms is $O(kN^2)$.

Algorithm 5 Merge bins in BRR-SS.**Input:** binsToMerge, numBins**Output:** Merged bins binsToMerge

```

1: mergedBins  $\leftarrow$  List of size numBins
2: startIndex  $\leftarrow \frac{1}{4} \cdot \text{numBins}$ 
3: endIndex  $\leftarrow \frac{3}{4} \cdot \text{numBins}$ 
4: for  $i = \text{startIndex} + 1$  to  $\text{endIndex}$  do
5:   firstToMerge  $\leftarrow 2 \cdot (i - \text{startIndex}) - 1$ 
6:   secondToMerge  $\leftarrow 2 \cdot (i - \text{startIndex})$ 
7:   mergedBins[i]  $\leftarrow$  binsToMerge[firstToMerge]
     + binsToMerge[secondToMerge]
8: end for
9: return mergedBins

```

D. Autoregressive First In, First Out-buffer

The second new algorithm resulting from the project, designated Autoregressive First In, First Out-buffer (AR-FIFO), autoregressively adds information to every sample in the cache. The algorithm is supervised and as such, also updates the corresponding sample target values \mathbf{C}_Y . Presented in Algorithm 6 is the pseudocode for AR-FIFO, where \mathbf{C}_X contains the feature values of samples in the cache, and \mathbf{C}_Y the target values.

Algorithm 6 Create cache of size N using AR-FIFO.**Input:** $\{(\mathbf{X}_1, \mathbf{Y}_1), (\mathbf{X}_2, \mathbf{Y}_2), \dots, (\mathbf{X}_t, \mathbf{Y}_t), \dots\}, a$ **Output:** Cache \mathbf{C}

```

1:  $\mathbf{C} \leftarrow$  Empty cache of size  $N$ 
2:  $t \leftarrow 1$ 
3: loop
4:    $\mathbf{X}_t, \mathbf{Y}_t \leftarrow$  Newly received sample
5:   if  $t \leq N$  then
6:      $\mathbf{C}_X[t] = \mathbf{X}_t$ 
7:      $\mathbf{C}_Y[t] = \mathbf{Y}_t$ 
8:   else
9:      $\mathbf{C}_X[0 : N-1] \leftarrow a\mathbf{C}_X[0 : N-1] + (1-a)\mathbf{C}_X[1 : N]$ 
10:     $\mathbf{C}_X[N] \leftarrow a\mathbf{C}_X[N] + (1-a)\mathbf{X}_t$ 
11:     $\mathbf{C}_Y[0 : N-1] \leftarrow a\mathbf{C}_Y[0 : N-1] + (1-a)\mathbf{C}_Y[1 : N]$ 
12:     $\mathbf{C}_Y[N] \leftarrow a\mathbf{C}_Y[N] + (1-a)\mathbf{Y}_t$ 
13:   end if
14:    $t \leftarrow t + 1$ 
15: end loop

```

Initially, the first N samples fill the cache. Then, every sample in the cache gets updated autoregressively. How much information is retained from the start of the monitoring system is regulated by the autoregression coefficient a . The complexity of multiplying a scalar with a matrix gives the complexity of AR-FIFO, resulting in $\mathcal{O}(kN)$. If $a = 0$, a newly received sample will be added to the end of the cache while the earliest sample located at the beginning will be removed. With $a = 1$, the cache retains the initial N samples without further processing. However, selecting a value a between 0 and 1 results in the retention of information from previous samples. As the choice of a affects the prediction accuracy, the resulting prediction accuracy will be provided for several values of a .

V. RESULTS

This section initially provides prediction accuracy plots for BRR-SS without model re-computation using different parameter values for the traces ‘KV flashcrowd - JNSM 2017’ and ‘VoD periodic - CNSM 2015.’ Afterward, the result for AR-FIFO is similarly presented with different autoregressive coefficient values a . Following the evaluation of varying parameter values are comparisons between RS, RR-SS, BRR-SS, and AR-FIFO while using the best-determined parameters for the latter two. Presented is the prediction accuracy without model re-computation, with periodic model re-computation and model re-computation using change detection for both traces. Additionally, an offline benchmark trained using 70% of the samples, randomly selected, of each respective trace with the test set being the remaining 30%, is also presented. Finally, shown is also the average time to process a received sample for different cache sizes on both traces for all investigated algorithms.

A. Prediction Accuracy for BRR-SS Using Different Parameter Values

Figures 4a and 4b show the NMAE of the two different traces using a varying number of bins for BRR-SS. Shown in Figures 4c and 4d is the NMAE of BRR-SS using 16 bins with different targets used during the binning process, that being $target$ in Algorithm 4. Compared are four features from the sorted list of the 16 most important features, where 1 is the most important and 16 the least. Included is also the use of the target value \mathbf{Y}_t , resulting in a supervised version of BRR-SS.

The bin count of BRR-SS has a relatively small impact on the NMAE as seen in Figure 4a, which exhibits the most conclusive results as the variance of Figure 4b is overall high. However, from a prediction accuracy standpoint, an optimal amount seems to be around 16 for smaller cache sizes such as $N = 32$ while for larger values of N , for example, $N = 2048$, the difference is within the margin of error as the 95% confidence intervals are intersecting. Since the results show the best overall prediction accuracy for 16 bins, all further evaluations will use 16 bins unless otherwise specified.

Much like for the number of bins, the use of a different $target$ for BRR-SS has an overall small impact on the NMAE as seen in Figures 4c and 4d. An exception is the target value \mathbf{Y}_t which seems to result in inconsistent behavior. For example in Figure 4c where \mathbf{Y}_t results in slightly lower NMAE for $N = 2048$ compared to the used features, while it for $N = 128$ and $N = 512$ is noticeably larger.

B. Prediction Accuracy for AR-FIFO Using Different Parameter Values

The prediction accuracy for AR-FIFO with different parameter values a is shown in Figure 5. Figures 5a and 5b show the NMAE for values of a ranging from 0 to 1 with a step size of 0.2 for the traces ‘KV flashcrowd - JNSM 2017’ and ‘VoD periodic - CNSM 2015,’ respectively. Furthermore, the prediction accuracy in finer step sizes around the best performing values of a is shown in Figures 5c and 5d again for each respective trace.

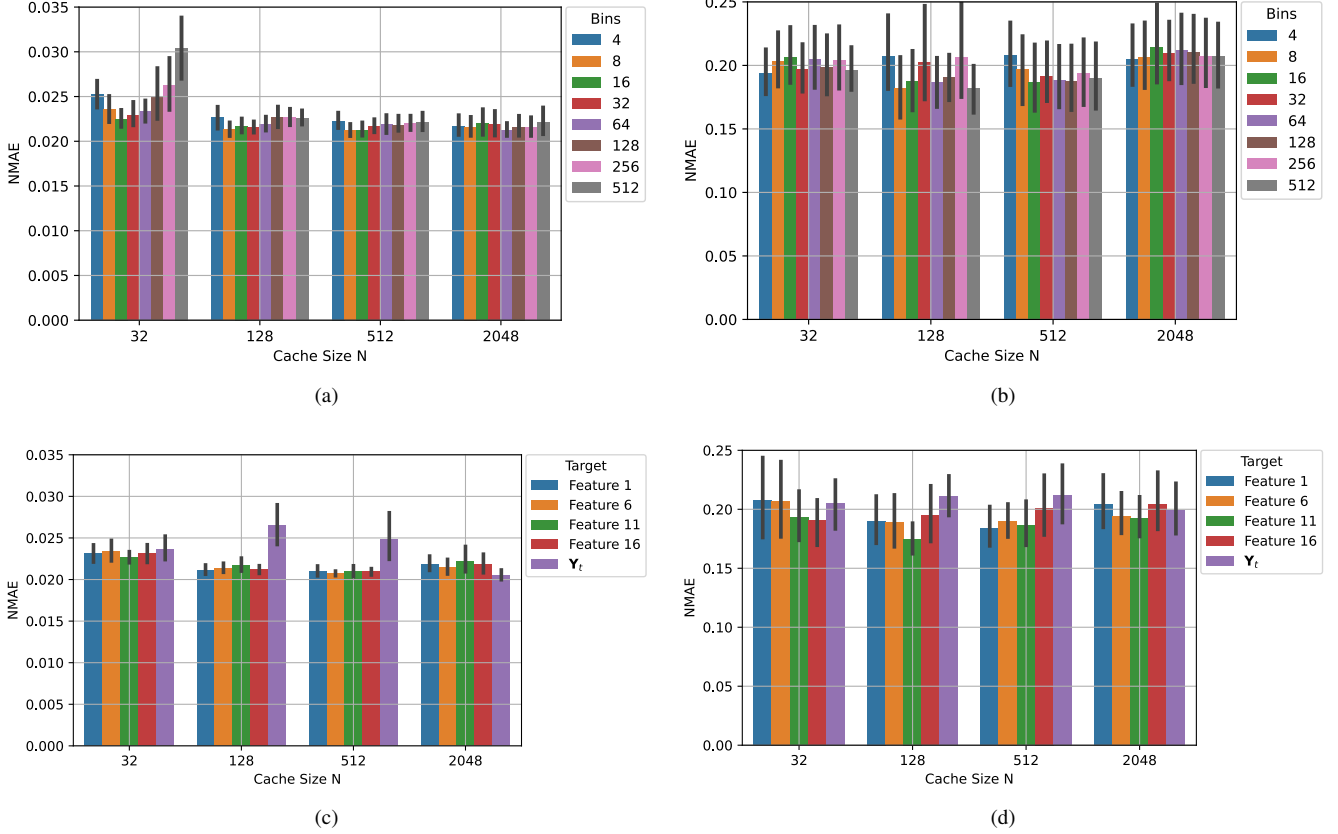


Fig. 4. NMAE of BRR-SS as a function of cache size N for different numbers of bins on the top row using the most important feature as *target* while on the bottom row using different features or labels and 16 bins. The “KV flashcrowd - JNSM 2017” trace is to the left, and the “VoD periodic - CNSM 2015” trace is to the right.

The results for the “KV flashcrowd - JNSM 2017” with a values ranging from 0 to 1 shown in Figure 5a indicate that a higher value for a below 1 produces the best prediction accuracy for the algorithm, especially for larger cache sizes. As a result, values above and below $a = 0.8$ were chosen for the finer step sizes in Figure 5c, resulting in values $a = 0.65, 0.70, \dots, 0.90, 0.95$ for that particular trace. Among those values, $a = 0.85$ and $a = 0.90$ have the best prediction accuracy for the most amount of cache sizes, although the NMAE value for all parameter values in the cache sizes 512 and 2048 lie within the same 95% confidence interval.

The results for the “VoD periodic - CNSM 2015” trace with values ranging from $a = 0$ and $a = 1$ are shown in Figure 5b. The values $a = 0$ and $a = 1$ have significantly better prediction accuracy than others. According to the algorithm, when $a = 0$ the algorithm adds the newest sample without retaining any information from previously removed samples, and for $a = 1$ it retains the initial values of the cache without ever updating it. The result from using finer step sizes of a is shown in Figure 5d. The prediction accuracy is however worse for those values of a when compared to $a = 0$ and $a = 1$.

Concluded from the given results, all further evaluations use $a = 0.85$ and $a = 0$ on the “KV flashcrowd - JNSM 2017” trace and “VoD periodic - CNSM 2015” trace, respectively.

C. Prediction Accuracy Without Model Re-computation

Figures 6a and 7a show the NMAE for the investigated traces without model re-computation. In other words, the evaluation is performed on all subsequent samples after the model is trained on samples selected from the first 3000 samples starting from t_0 .

As seen from the prediction accuracies on the “KV flashcrowd - JNSM 2017” trace, BRR-SS consistently matches or outperforms RS and RR-SS in Figure 6a, where the latter has a considerably higher NMAE for $N \leq 512$. AR-FIFO results in prediction accuracy on par with RR-SS for $N = 32$ while outperforming the offline benchmark for $N = 2048$ on the trace.

The resulting prediction accuracies on the “VoD periodic - CNSM 2015” trace found in Figure 7a are generally more inconclusive. However, AR-FIFO seems to perform noticeably worse overall.

D. Prediction Accuracy With Periodic Model Re-computation

Figures 6c-6g and 7c-7g show the NMAE while using periodic model re-computation with an interval of T_c samples. Figures 6c-6g show the most conclusive results as the different algorithms differ more in their prediction accuracy compared to the “VoD periodic - CNSM 2015” trace in Figures 7c-7g. The improvement in variance is overall significant when

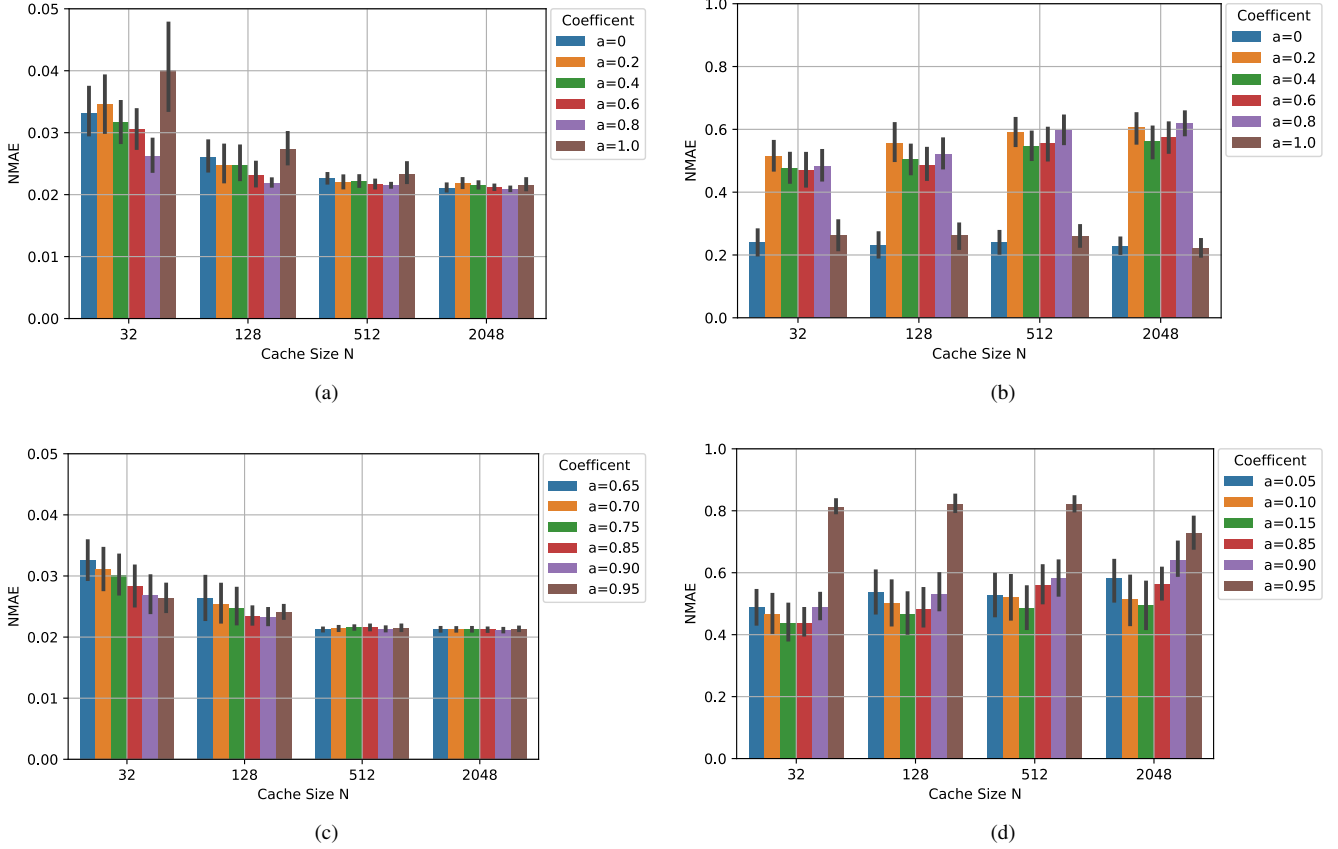


Fig. 5. NMAE of AR-FIFO as a function of cache size N where the different bars correspond to different values for the autoregressive coefficient a . The “KV flashcrowd - JNSM 2017” trace is to the left, and the “VoD periodic - CNSM 2015” trace is to the right.

compared to no model re-computation, gradually becoming smaller as T_c decreases, seen on both traces. Additionally, an improvement in NMAE can in multiple cases also be seen, especially evident for AR-FIFO in Figures 7c-7g, but also in the case of BRR-SS and RS in Figures 6c and 6d, where the former is very close to the offline benchmark for $N = 512$ and $N = 2048$.

RR-SS, much like AR-FIFO, in Figure 6 has a significantly lowered variance while using periodic model re-computation compared to no model re-computation. However, RR-SS seems to have a noticeably higher NMAE for $N = 2048$, occurring on both traces while using periodic model re-computation.

E. Prediction Accuracy With Model Re-computation Using Change Detection

Tables I and II show the number of model re-computations performed using the change detection method STUDD for each respective investigated trace from a specific randomly chosen starting point t_0 . For comparison, the number of model re-computations performed using periodic model re-computation with $T_c = 100$ and $T_c = 2000$ is also seen. The number of model re-computations performed using change detection compared to that of periodic model re-computation varies for the different traces. In Table I, change detection has fewer model re-computations than for any of the investigated

TABLE I
THE NUMBER OF MODEL RE-COMPUTATIONS BASED ON STARTING POINT t_0 ON THE “KV FLASHCROWD - JNSM 2017” TRACE OF LENGTH 19334.

t_0	Change detection	Periodic $T_c = 100$	Periodic $T_c = 2000$
3372	3	159	7
3454	3	158	7
3668	2	156	7
6148	3	131	6
6209	6	131	6
8509	3	108	5
10287	3	90	4
11444	2	78	3
14544	1	47	2
14644	1	46	2

values of T_c , around half of $T_c = 2000$ while it in Table II falls somewhere in between $T_c = 100$ and $T_c = 2000$. The difference in the number of model re-computations indicates that it better adapts to the given trace by adjusting the frequency at which the model is updated, for example by adapting to the periodicity of the “VoD periodic - CNSM 2015” trace visualized in Figure 1b.

Figures 6b and 7b show the prediction accuracy when using change detection for each respective trace. The NMAE for change detection on the “KV flashcrowd - JNSM 2017” trace in Figure 6b follows a similar trend to the results found without model re-computation. The prediction accuracy of

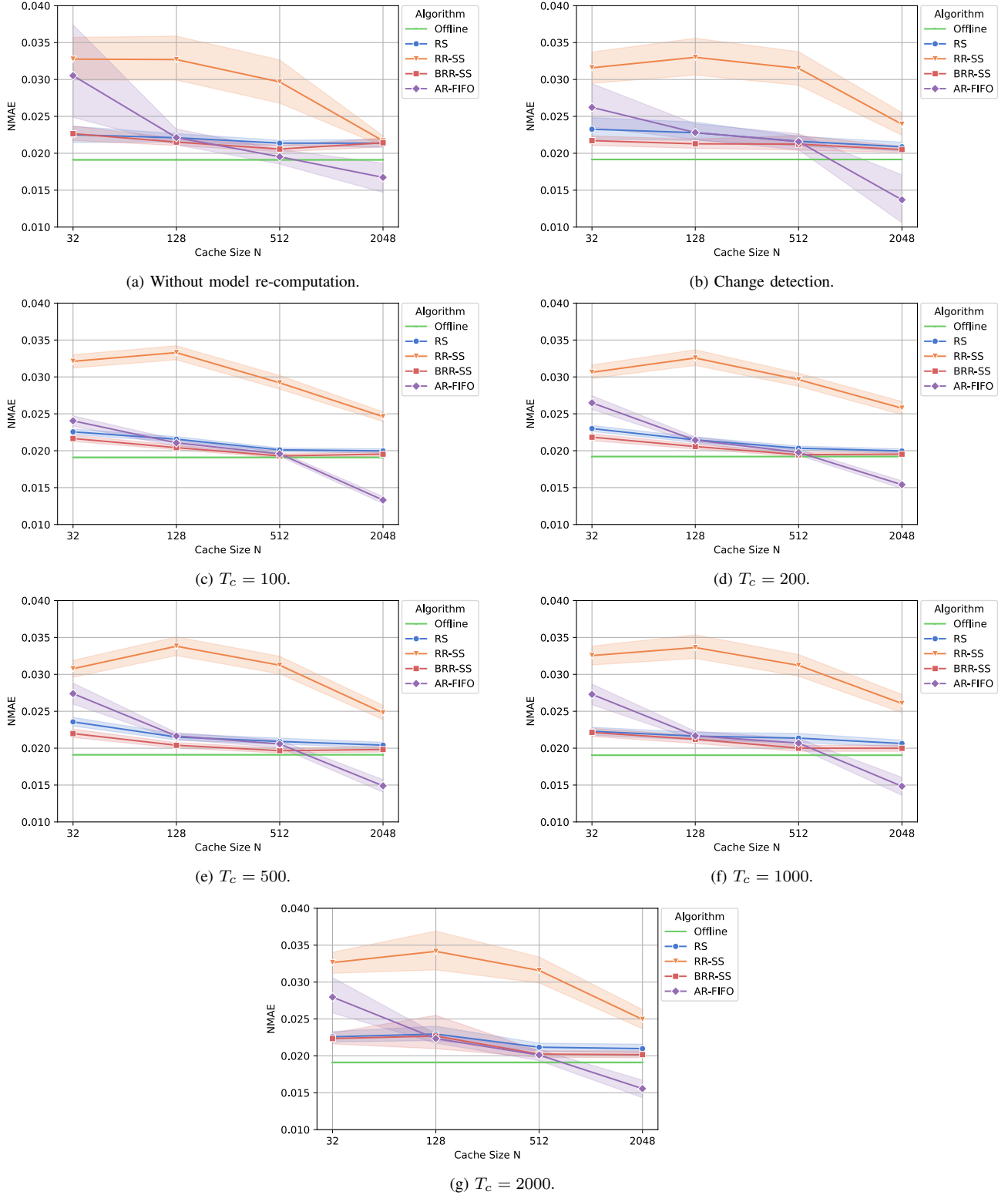
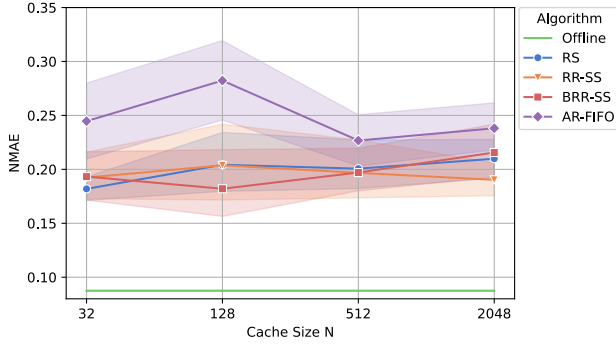


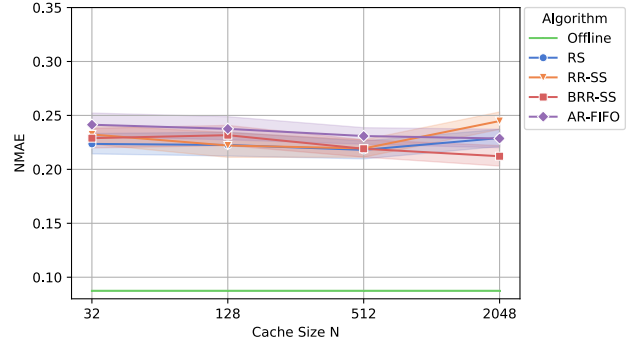
Fig. 6. NMAE as a function of cache size N on the “KV flashcrowd - JNSM 2017” trace without model re-computation, with periodic model re-computation for different T_c , and model re-computation using change detection. BRR-SS uses 16 bins, and *target* is the most important feature, while AR-FIFO uses $a = 0.85$.

RS improves slightly with cache size but remains largely unchanged. RR-SS has the worst prediction accuracy with a wide margin for all cache sizes except $N = 2048$, where it

approaches the NMAE value of RS. BRR-SS outperforms RS for all cache sizes, however, the NMAE for both algorithms lies within the same confidence interval for $N = 512$ and



(a) Without model re-computation.



(b) Change detection.

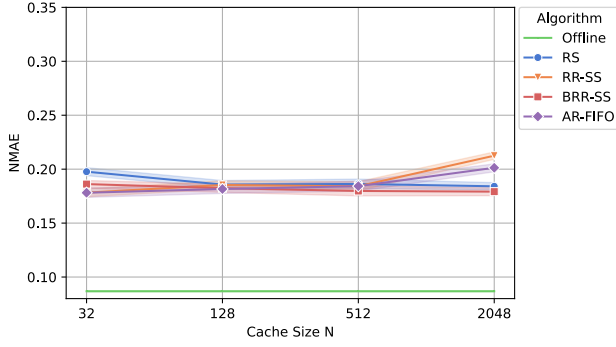
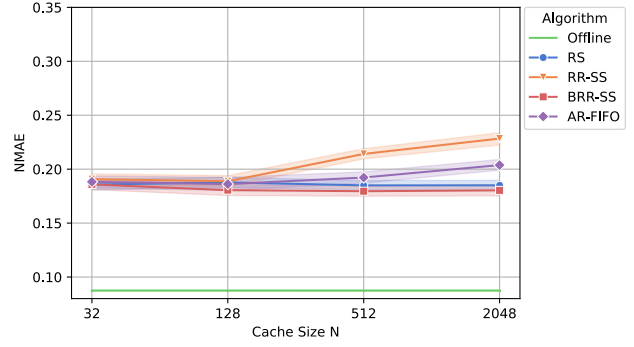
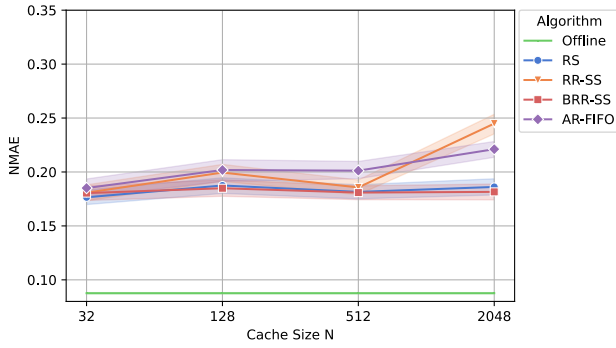
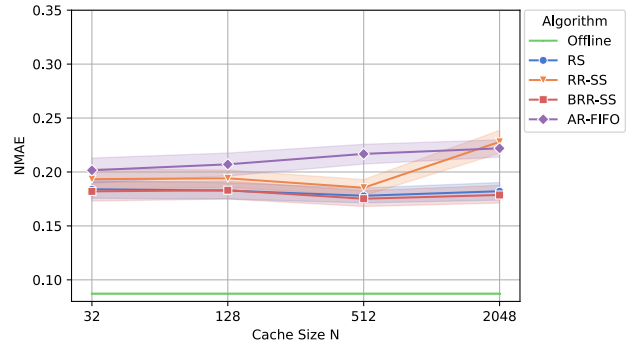
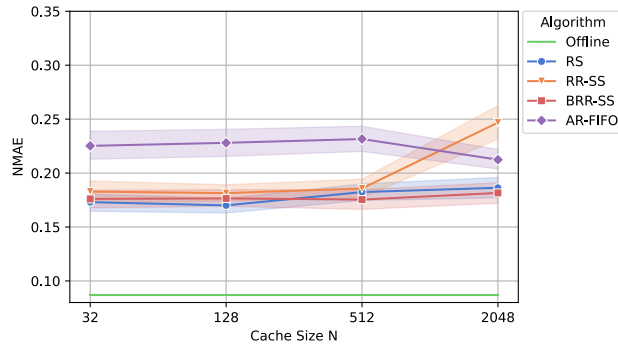
(c) $T_c = 100$.(d) $T_c = 200$.(e) $T_c = 500$.(f) $T_c = 1000$.(g) $T_c = 2000$.

Fig. 7. NMAE as a function of cache size N on the “VoD periodic - CNSM 2015” trace without model re-computation, with periodic model re-computation for different T_c , and model re-computation using change detection. BRR-SS uses 16 bins, and *target* is the most important feature, while AR-FIFO uses $a = 0$.

$N = 2048$. AR-FIFO performs worse than RS for $N = 32$, but matches it for $N = 128$ and $N = 512$, and outperforms the offline benchmark for cache size $N = 2048$.

The prediction accuracy when using change detection on the “VoD Periodic - CNSM 2015” trace is shown in Figure 7b. RS, RR-SS, and BRR-SS have similar prediction accuracies,

with potential outliers being BRR-SS having slightly higher NMAE for $N = 128$ but the best prediction accuracy for $N = 2048$ and RR-SS having the worst overall NMAE for $N = 2048$. AR-FIFO has a slightly worse prediction accuracy when compared to the other algorithms for smaller cache sizes but matches RS for cache size $N = 2048$.

Comparing periodic model re-computation and change detection on the “KV flashcrowd - JNSM 2017” trace in Figures 6b-6g, change detection seems favorable in terms of prediction accuracy as Figure 6b has similar variance and NMAE as periodic model re-computation using $T_c = 2000$ in Figure 6g while only using around half the number of model re-computations. Change detection is, however, close to that of using no model re-computation in Figure 6a. Table II shows that the number of model re-computations on the “VoD Periodic - CNSM 2015” trace while using change detection is roughly that of $T_c = 200$ or $T_c = 500$. However, the comparison does not translate to prediction accuracy as even the overall prediction accuracy of $T_c = 2000$ with far fewer model re-computations is much better. Only AR-FIFO for larger values of T_c is comparable between the two methods, where the remaining algorithms perform significantly better using periodic model re-computation, a potential exception being $N = 2048$ for RR-SS. The variance while using change detection in Figure 7b is greatly improved over no model re-computation in Figure 7a. However, the NMAE seems to, outside of AR-FIFO, be overall worse.

The higher NMAE of RR-SS for $N = 2048$ for both periodic model re-computation and change detection in Figures 6b-6g and 7b-7g, when compared to no model re-computation, could be caused by issues in the distribution becoming greater as more samples are processed. Contrarily to RR-SS, algorithms maintaining distribution such as RS and BRR-SS seem to benefit from the model re-computation for larger cache sizes as their NMAEs are close to that of the offline benchmark, especially in Figures 6c-6f. AR-FIFO benefits the most overall, however, particularly in Figures 7b-7g but also in Figures 6b-6g, where a significant reduction in variance and NMAE can be seen, especially for $N = 32$ and $N = 2048$.

F. Computational Overhead of the Sample Selection Algorithms

Since fixed cache sizes are used, the computational overhead of the sample selection algorithms is primarily the time of processing a newly received sample. Figures 8 and 9 show the time to process a sample for each algorithm across 20 runs, including different bin counts for BRR-SS. The calculations were performed using an AMD Ryzen 7 5800X running at 4.7 GHz utilizing 4 of its 16 threads.

As seen in Figures 8 and 9, representing each respective trace, RS has the lowest computational overhead for all cache sizes N . AR-FIFO is trailing behind RS, but with better scaling for larger cache sizes as its processing time does not increase as rapidly.

BRR-SS has a significantly higher process time when compared to RS and AR-FIFO as it uses a relevance and a redundancy list, making it more computationally heavy. If

TABLE II
THE NUMBER OF MODEL RE-COMPUTATIONS BASED ON STARTING POINT t_0 ON THE “VoD PERIODIC - CNSM 2015” TRACE OF LENGTH 48292.

t_0	Change detection	Periodic	Periodic
		$T_c = 100$	$T_c = 2000$
4728	47	435	21
9720	36	385	19
11205	28	370	18
12910	43	353	17
16120	48	321	16
25704	31	225	11
29878	24	181	9
36911	11	113	5
42862	9	54	2
45216	1	30	1

compared to RR-SS, however, BRR-SS can be seen consistently outperforming RR-SS while using 16 bins except for $N = 2048$ in Figure 8a. However, not the case in Figure 9a, where the time to process a sample for RR-SS with $N = 2048$ is higher when compared to BRR-SS. The use of more bins for BRR-SS does, as expected, reduce the time to process a sample, especially for larger values such as $N = 2048$, where the reduction can be significant, as seen in Figures 8b and 9b.

VI. DISCUSSION

A. Importance of Model Re-computation

The possibility of updating the prediction model more frequently for improved flexibility is a strong argument for online learning. For example, periodic model re-computation consistently lowers the variance of the NMAE but can also improve error rates as the model is more frequently updated to represent the current state of the trace more accurately.

The results found regarding change detection indicate that periodic model re-computation could be the preferred method as there does not seem to be a strong case for change detection regarding prediction accuracy on the investigated traces. Two aspects in favor of periodic model re-computation are that it does not require the knowledge of, in this case, the 1000 latest samples but also the reduced computational overhead by not using a teacher and student model. However, an aspect in favor of change detection is that one does not need to find an optimal value for T_c for the given trace, as change detection can better adapt to the trace. One can, however, argue that the single parameter T_c of periodic model re-computation makes it less complicated than STUDD with its multiple ones.

B. Prediction Accuracy of RS and RR-SS

RS has consistently shown low error rates, indicating that maintaining good distribution could be a dominant aspect in the prediction accuracy of sample selection algorithms. However, removing deviating samples during preprocessing increases the overall sample quality, benefiting the algorithm as it only selects randomly. Therefore, in a real-world application, the algorithm may perform worse.

RR-SS should not be affected as much as RS if preprocessing is omitted since samples of poor perceived quality should be excluded, as it selects samples based on quality. The cause

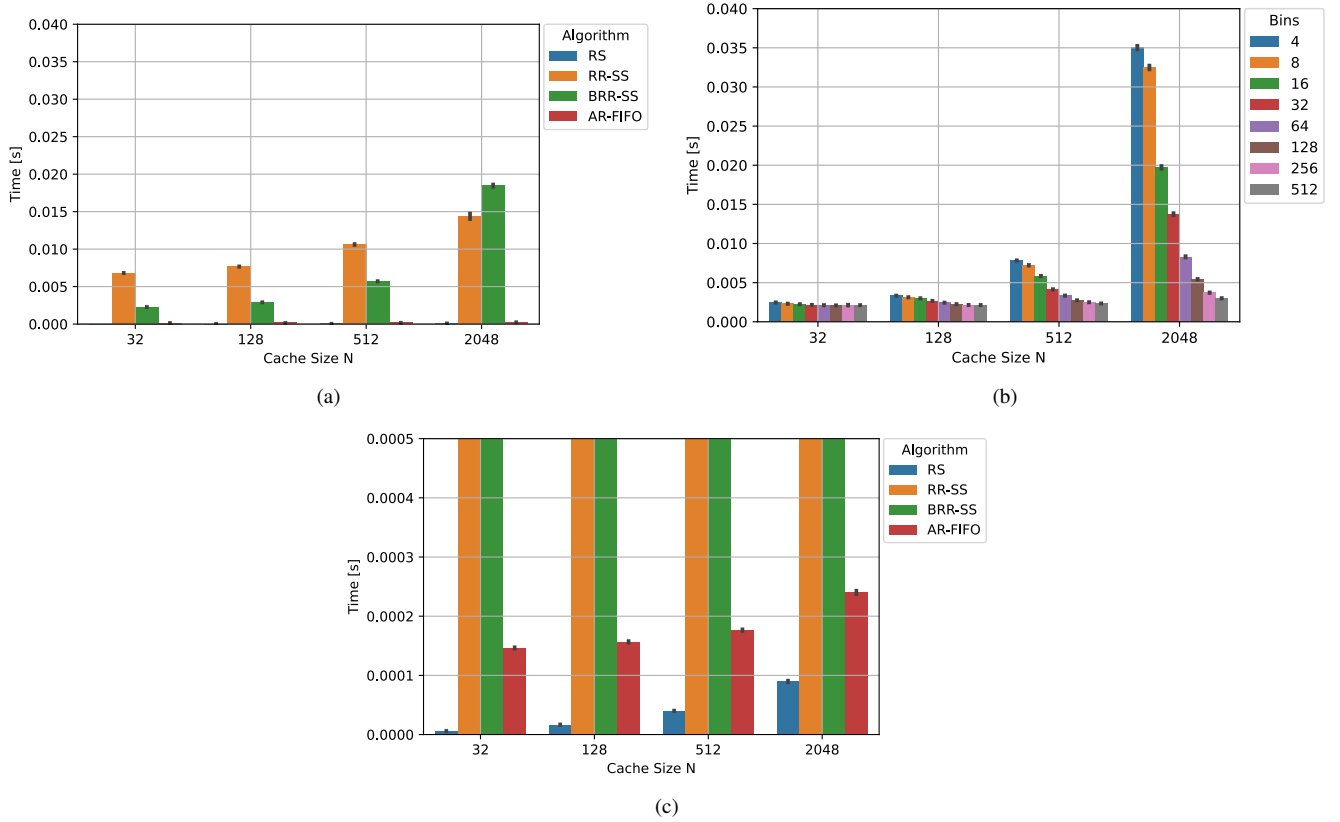


Fig. 8. Time to process a sample as a function of cache size N on the “KV flashcrowd - JNSM 2017” trace. BRR-SS uses 16 bins in the top left and different amounts of bins in the top right, while AR-FIFO always uses $a = 0.85$. Zoomed in version of the top left figure is found on the bottom row.

of the overall high NMAE of RR-SS on the investigated traces is most likely the lack of distribution found in the algorithm. An attribute less impactful for larger cache sizes where there are enough samples to cover all ranges of values, hence its consistently improved prediction accuracy for $N = 2048$ compared to smaller cache sizes on the “KV flashcrowd - JNSM 2017” trace in Figure 6. It is, however, worth noting that the opposite is generally true on the “VoD periodic - CNSM 2015” trace in Figure 7 which has fewer possible target values, making the exclusion of a range of target values less likely.

C. Prediction Accuracy of BRR-SS

Further development on BRR-SS can be of interest as it shows overall consistent and promising results. The robustness of having the positive aspects of both RS and RR-SS should mean that its relative prediction accuracy is consistent across different traces and applications.

The difference in prediction accuracy when using different amounts of bins seen in Figure 4a indicates that the largest impact is seen for smaller cache sizes N . The decreased prediction accuracy while using fewer bins, such as 4, is most likely attributed to the increased importance of maintaining distribution for smaller N . Distribution becomes important for smaller N as there are not enough samples to automatically cover all ranges of target values, an effect similar to that seen for RR-SS. Using more bins seems to have the opposite effect, where the amount, such as 512, becomes excessive

compared to the cache size. The negative impact of using an excessive amount of bins could come down to inefficiencies in calculating the bin rank list since each bin contains only a few samples.

Using a different value for $target$ during the binning process results in surprising outcomes. The use of Y_t should yield the most optimal binning. However, a conflict in the binning process where the relevance or redundancy list does not include the value used to determine which bin each sample belongs to, Y_t in this case, could be the cause of the inconsistent behavior seen in Figure 4c.

D. Prediction Accuracy of AR-FIFO

AR-FIFO has less consistent behavior across the different traces when compared to BRR-SS. However, for the right application, the algorithm can lead to the best results and even outperform the offline benchmark seen in Figure 6.

The improved prediction accuracy over the offline benchmark could be attributed to it not storing the samples themselves but instead data with an attribute of inertia, containing information from multiple past samples. This data seems to be better for predicting Y when compared to only using previous samples.

However, the worsened results found in Figure 7 indicate that the algorithm is not suited for all types of applications. These results are most likely caused by the algorithm simply functioning as a queue since $a = 0$. Using $a = 0$ results

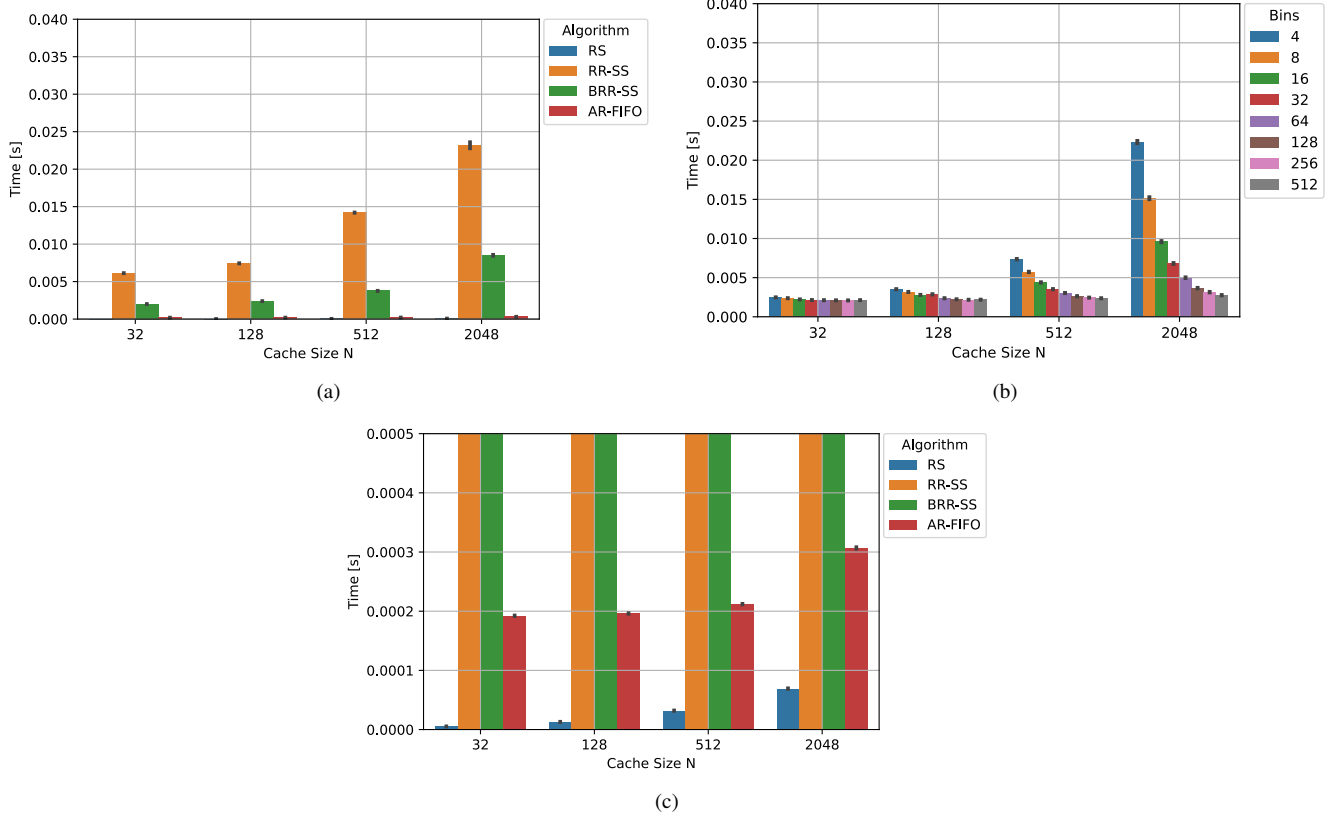


Fig. 9. Time to process a sample as a function of cache size N on the “VoD periodic - CNSM 2015” trace. BRR-SS uses 16 bins in the top left and different amounts of bins in the top right, while AR-FIFO always uses $a = 0$. Zoomed in version of the top left figure is found on the bottom row.

in the lowest NMAE on the “VoD periodic - CNSM 2015” trace as inertia in the cache does not seem to suit the discrete nature of the target values seen in Figure 1b. The opposite is the case for the more continuous target values on the “KV flashcrowd - JNSM 2017” trace in Figure 1a, where inertia shows promising results.

However, AR-FIFO improves significantly with the use of more frequent model re-computations on the “VoD periodic - CNSM 2015” trace in Figure 7. Since $a = 0$, the cache contains the N most recent samples, resulting in the prediction accuracy relying on having the most up-to-date samples instead of, for example, using those of the highest quality, resulting in significant improvements by updating the model more frequently.

E. Computational Overhead of Algorithms

Achieving low computational overhead is a fundamental aspect of online sample selection algorithms as online learning frequently receives new samples that have to be processed. However, as long as the time to process a sample is shorter than the interval between receiving samples, one second in the case of the investigated traces, the difference should be minimal. The fixed time between samples enables algorithms such as RR-SS and BRR-SS to have viable applications where the interval is large enough.

The computational overhead for the different algorithms is, overall, as expected, where RS essentially becomes a bench-

mark as it is difficult to improve upon its low computational overhead. The higher probability of making changes in the cache of RS as N increases is probably the cause of the increasing processing time for larger cache sizes seen in Figures 8c and 9c. An argument for AR-FIFO scaling better for larger cache sizes can be made. However, this could come down to implementation, especially considering that the complexity of RS is $\mathcal{O}(1)$ while it for AR-FIFO is $\mathcal{O}(kN)$.

RR-SS outperforming BRR-SS in computational overhead for $N = 2048$ in Figure 8a could be caused by RR-SS not replacing a sample in the cache if the average rank is high enough. Resulting in the calculation of `cacheRankList` being frequently omitted hence reducing the processing time. However, BRR-SS does have the flexibility of changing its number of bins to acquire a lower sample processing time. The overall lower computational overhead of BRR-SS compared to RR-SS, despite the same complexity, is most likely attributed to it not having to calculate `rankNew` and `rankAvgSample`, but also the fewer number of samples included in `binRankList` in the former compared to `cacheRankList` in the latter.

A deeper analysis can be made when looking at the different number of bins for BRR-SS in Figures 8b and 9b. As expected, using more bins will reduce the time of processing a sample, most noticeable for larger cache sizes N , where the number of samples included in the selected bin becomes significantly lower. The greatest downside of using a large number of bins is

the reduced prediction accuracy for smaller cache sizes such as $N = 32$ and $N = 128$ seen in Figure 4a. However, the impact on prediction accuracy is minimal for larger cache sizes such as $N = 512$ and $N = 2048$, where the time to process a sample is drastically lowered. In other words, as the cache size increases, so should the number of bins to achieve a more similar prediction accuracy and computational overhead across all cache sizes N .

F. Comparing Online and Offline Learning

In general, the prediction accuracy of the offline benchmark is better than that of online learning on the investigated traces seen in Figures 6 and 7. However, offline learning can be problematic in realistic situations where gathering and storing enough data ahead of time might be impractical. Continuously updating the cache and model using online learning can therefore be used to improve flexibility and adaptability. As a side note, since the offline benchmark selects its training set from randomly selected samples across the entire trace, it might not represent the same result as a training set consisting of the first 70% of samples. Included in the training set of the offline benchmark are any potential overarching changes in the trace, which might not reflect a realistic scenario where concept drift may occur.

Found in Figure 7 is the largest difference between online learning and offline learning on the investigated traces, where the NMAE of the former is around twice as large as that of the latter for $N = 32$. However, considering that online learning uses only 32 samples while offline learning around 33804 to train the prediction model, the difference in prediction accuracy can be deemed acceptable. In some cases, such as Figure 6, online learning can match or even outperform offline learning despite using significantly smaller data sets, even though the offline benchmark in this case only uses around 13534 samples for its training set.

Periodic model re-computation, for example, can significantly improve the prediction accuracy of online learning where even $T_c = 100$ can be manageable with such small cache sizes. Furthermore, both model re-computation and processing of the latest sample could potentially be performed between receiving samples, avoiding potential disruption in received samples if there are restrictions on whether or not a sample is processable while the model is being re-computed.

The difference in prediction accuracy among different cache sizes is, in general, surprisingly small. There are exceptions such as RR-SS and AR-FIFO in Figure 6 benefiting from larger cache sizes. However, Figure 7 shows overall smaller variations among different values of N , where smaller values in multiple cases even outperform larger ones. Depending on the situation, the potential decrease in prediction accuracy can be deemed acceptable, for example between $N = 32$ and $N = 2048$, considering the reduced computational overhead by using only $\frac{1}{64}$ of the number of samples.

G. Future Work

As seen, some algorithms such as AR-FIFO can achieve outstanding prediction accuracy during specific load patterns

or prediction targets on the investigated traces. Therefore, investigating methods that can detect and dynamically change the used sample selection algorithm and its parameters to accommodate the used trace could be of great value to better generalize the use of online learning methods for different applications. The previously mentioned clothes sales analogy highlights that trends, or in the case of networked system load patterns, can vary over time. Another example of varying demands is the power grid load surge known as “TV Pickup.” During commercials for prime time television in the United Kingdom, a large number of television watchers take a break and turn on kettles and open refrigerators at the same time, which according to [11] has power usage surges of over 2000 MW on record. Similar to the power usage, “Video on Demand” services have a different amount of users distributed across a day, not to mention how the release of a season finale could affect a streaming service. Evaluating the prediction accuracy for an entire day with only a single sample selection algorithm and comparing it to switching between different algorithms during different load patterns could be a relevant field of study.

Furthermore, investigating different selection methods using the binning method found in BRR-SS could be of interest as the algorithm is only one implementation of the concept. Additionally, left as future work are more optimized versions of all investigated algorithms.

VII. CONCLUSION

This project has resulted in two new methods for selecting samples, BRR-SS and AR-FIFO, aimed toward online learning. The former for more general applications with overall favorable prediction accuracy compared to existing investigated sample selection algorithms on the investigated traces. The latter focuses on more specialized applications with a prediction accuracy that can significantly outperform even an offline benchmark while maintaining minimal computational overhead. Both algorithms have different approaches resulting in the proposal of distinct concepts with varying properties such as applications and computational overhead.

Furthermore, different types of model re-computation methods have been investigated, showing how the significantly lower computational overhead of online learning in multiple scenarios can match the prediction accuracy of offline learning. Preferred over the use of change detection on the investigated traces is the simplicity of periodic model re-computation. Traces with discrete target values, for example, the “VoD periodic - CNSM 2015” trace, have shown greater difficulty in prediction accuracy for online learning. However, traces of a more continuous nature, such as the “KV flashcrowd - JNSM 2017” trace, shows especially promising results in favor of online learning.

APPENDIX A

SORTED LIST OF MOST IMPORTANT FEATURES

ACKNOWLEDGMENT

The authors would like to thank Rolf Stadler for making the project possible and Xiaoxuan Wang for her dedication,

guidance, and help by holding weekly meetings and giving valuable feedback on the project work.

REFERENCES

- [1] A. Kullen, “Projektvalskatalog - EF112X kandidatexamensarbete inom elektroteknik (15 hp) våren 2022,” handout in the course *EF112X Examensarbete inom elektroteknik, grundnivå*, KTH, Stockholm, Sweden, 2022.
- [2] R. S. Villaça and R. Stadler, “Online learning under resource constraints,” in *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2021, pp. 134–142.
- [3] S. Agrahari and A. K. Singh, “Concept drift detection in data stream mining : A literature review,” *Journal of King Saud University - Computer and Information Sciences*, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1319157821003062>
- [4] (2020, Jun.) Data traces for “Efficient learning on high-dimensional operational data” paper, CNSM 2019. KTH. [Online]. Available: <https://github.com/foroughsh/KTH-traces>
- [5] 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.
- [6] R. Stadler and X. Wang, “Online learning with sample selection - Part I,” handout of project tasks from supervisors in the course *EF112X Examensarbete inom elektroteknik, grundnivå*, KTH, Stockholm, Sweden, 2022.
- [7] V. Cerqueira, H. M. Gomes, A. Bifet, and L. Torgo, “STUDD: A student-teacher method for unsupervised concept drift detection,” *CoRR*, vol. abs/2103.00903, 2021. [Online]. Available: <https://arxiv.org/abs/2103.00903>
- [8] R. Stadler and X. Wang, “Online learning with sample selection - Part II,” handout of project tasks from supervisors in the course *EF112X Examensarbete inom elektroteknik, grundnivå*, KTH, Stockholm, Sweden, 2022.
- [9] E. S. Page, “Continuous Inspection Schemes,” *Biometrika*, vol. 41, no. 1-2, pp. 100–115, 06 1954. [Online]. Available: <https://doi.org/10.1093/biomet/41.1-2.100>
- [10] C. C. Aggarwal, *Data Mining: The Textbook*. New York: Springer, 2015, vol. 1, ch. 2.4.1.2 Reservoir Sampling for Data Streams, pp. 39–40.
- [11] (2007, Sep.) Can you have a big ‘switch off’? BBC. London, United Kingdom. [Online]. Available: <http://news.bbc.co.uk/1/hi/magazine/6981356.stm>

Cybersecurity Evaluation of an IP Camera

Tova Stroeven and Felix Söderman

Abstract—The prevalence of affordable internet-connected cameras has provided many with new possibilities, including keeping a watchful eye on property and family members from afar. In order to avoid serious breaches of privacy, it is necessary to consider whether these devices are secure. This project aims to evaluate the cybersecurity of one such device, an IP camera from Biltema. This was done by performing an extensive analysis of the camera, determining possible vulnerabilities, and performing penetration tests based on identified vulnerabilities. The tests included capturing and analyzing network traffic, attempting to crack the camera credentials, and attempting to disable the camera completely. The conclusions were that the camera should not be used for any security applications and is unsuitable to use in situations where one's privacy is important.

Sammanfattning—Det breda utbudet av prisvärda och kameror med internet uppkoppling har medfört helt nya möjligheter. Idag är det till exempel möjligt att hålla koll på sina barn utan att vara i rummet, eller hålla ett öga på hemmet via mobilen. Det är dock nödvändigt att reflektera över om dessa enheter är säkra, för att undvika allvarliga integritetsintrång. Projektets syfte är att utvärdera cybersäkerheten hos en sådan enhet, en IP-kamera från Biltema. Utvärderingen bestod av en omfattande analys av kameran, identifikation av möjliga sårbarheter och utförande av ett antal penetrationstester baserat på de upptäckta sårbarheterna. Testerna omfattade en analys av nätverkstrafik, att försöka knäcka kamerans inloggningssuppgifter samt att försöka inaktivera kameran. Slutsatsen var att kameran inte bör användas inom säkerhetstillämpningar och att den är olämplig i situationer där integritet är viktigt.

Index Terms—cybersecurity, IP camera, baby camera, security camera, penetration testing, IoT, Biltema, privacy, ethical hacking

Supervisor: Pontus Johnson

TRITA number: TRITA-EECS-EX-2022:165

I. INTRODUCTION

We live in a society where computers and technology constitute an important part of our everyday life, and more devices are connected to the Internet every day. These devices create what is known as the *Internet Of Things* (IoT). When developing these devices, price and new features take priority, while security aspects are often overlooked. This is because security research is time- and resource intensive, and is often not perceived as lucrative.

The lack of security awareness has flooded the market with a sea of insecure devices that create opportunity for hackers with malicious intent to infringe on our privacy. Due to this, it is crucial that the security of these devices are evaluated, their vulnerabilities mitigated, and that results are made public.

The integrity of IP cameras are particularly interesting to evaluate, since they pose a significant risk by having access to sensitive information. A compromised camera could mean a breach of privacy similarly to a home intrusion.

This study examines an IP camera purchased at a large Swedish retailer, to determine how secure the camera is from a cybersecurity perspective.

II. BACKGROUND

A. Selection of system

When selecting which system to evaluate, several factors were considered. The priority was to choose a system where the impact of a successful attack would be significant. The system chosen, an IP- and baby camera from Biltema, fulfilled this since it could mean a serious breach of privacy if an unauthorized external attacker would get access to the camera. Furthermore, if the camera is used for home surveillance, an attacker that manages to disable the camera would be able to enter without the risk of detection. The marketing of the camera did not advertise special security features, as opposed to similar cameras from other brands, which made this camera an interesting subject to examine. A mobile application by Biltema is used to view the camera feed and control the camera. The application could be a possible attack surface, especially since the last update was released in 2012. Additionally, the camera was being sold by a large retailer and has therefore been readily available to consumers, and at a reasonable price. Based upon all these factors, the IP- and baby camera from Biltema was chosen.

B. Related Work

IoT devices are common targets of cybersecurity evaluations. Since vulnerabilities often affect many different types of devices, organizations like Open Web Application Security Project (OWASP) compile lists of commonly occurring vulnerabilities, one example being *OWASP Top 10* [1] and another *OWASP IoT Top 10* [2]. Most of the common vulnerabilities described in [1], [2] have been explored in this report. Some vulnerabilities were not applicable to the system under consideration, and were thus excluded. For example, no Cross-site request forgery (CSRF) attacks could be attempted, since the web server was exposed through indirect object referencing (see section III-D2 for details). This means that a CSRF-attack could not be performed, since there is no protection to bypass.

IP-cameras are common targets of cybersecurity evaluations. The specific device considered in this project has not been previously tested. Because vulnerabilities may be similar across devices, it can be worthwhile to examine previous work regarding other IP-cameras. There are examples of cameras with a relatively high level of security, such as the ones evaluated in [3], [4]. On the contrary, there are IP cameras with several vulnerabilities [5], [6]. Often, these cameras are equipped with web servers, which provide an attacker with a large attack surface and thus makes it more likely that the device can be successfully hacked. [5], [6].

C. Delimitations

To properly evaluate the security of the IP-camera, several penetration tests of relevant security threats should be conducted. From [1], [2] it is clear that tests aimed at the cloud server could be highly relevant. However, penetration tests that attack the cloud connecting all cameras, and not just the device itself, could be legally problematic. These tests are therefore only discussed, and were not conducted. Another attack suggested in [1], [2] is CSRF. As discussed in section II-B, a CSRF-attack was not attempted.

Furthermore, the mobile application for the iPhone Operating System (iOS) has not been considered as a possible attack surface, because only Android devices were available.

It is necessary to consider as many relevant attacks and threats as possible, to be able to draw robust conclusions from penetration tests. However, time is a factor that needs to be considered, since it limits how many tests are feasible to perform within the time frame of the project, which in this case is a 15 ECTS bachelor's thesis. Due to this limited time, not all applicable vulnerabilities listed in *OWASP Top 10* could be considered. Server-Side Request Forgery was excluded both because it was at the bottom of the list, but also since it had a relatively low incidence rate according to OWASP [1]. The remaining *OWASP Top 10* vulnerabilities have been considered in some capacity, along with vulnerabilities related to the specific system under consideration.

The IP-camera was marketed to function with both Wi-Fi and Ethernet. However, during initial testing of the system, the camera only worked while connected using an Ethernet cable. Hence, all evaluations and penetration tests are conducted using Ethernet.

III. THEORY

This section contains descriptions of relevant tools, technologies, and attacks that are referred to in this report.

A. Tools

1) *Android 86x*: Android 86x is an open source project that ports the Android operating system to Intel x86 architecture [7]. This allows users to run Android on regular computer hardware or through virtualization technologies such as Oracle VM VirtualBox (see section III-A7).

2) *CyberChef*: CyberChef is a web application tool useful for data analysis. It has capability to encode, encrypt, and compress data. [8].

3) *decompiler.com*: www.decompiler.com is an online Android decompiler that converts APK files to java code [9].

4) *Ghidra*: Ghidra is a reverse engineering tool that can decompile different types of software. The program is developed by the American National Security Agency (NSA) [10], [11]. It is used to decode parts of the mobile application in this project, specifically decompiling .so-files into .c-files.

5) *Hydra*: Hydra is a parallelized network login cracker [12]. The program is open source and can be used for performing brute-force attacks (see section III-C2) and dictionary attacks (see section III-C5) on common network protocols such as Telnet (see section III-B5), SSH (Secure Shell) and

FTP (File Transfer Protocol). For this project, Hydra was used to target a Telnet connection.

6) *NMAP*: NMAP is a network tool used to scan for open ports on a network, and gain information regarding what services they most probably are running [13].

7) *Oracle VM VirtualBox*: Oracle VM VirtualBox is a virtualization software that allows the user to run multiple instances of other operating systems inside their already existing operating system [14].

8) *OWASP Threat Dragon*: OWASP Threat Dragon is an open source software made for creating threat model diagrams [15].

9) *PlayCap*: PlayCap [16] is a software used to play back network traffic captured by a program such as Wireshark (see section III-A10).

10) *Wireshark*: Wireshark is an application that is used to capture and analyze network traffic [17].

B. Protocols

1) *Ethernet / IEEE 802.3*: IEEE 802.3 is an IP network protocol used to provide a network connection over a cable, which is commonly referred to as "Ethernet" [18].

2) *HTTP*: Hypertext Transfer Protocol, or HTTP, is an unencrypted application layer protocol most commonly used to supply a client with HTML websites [19]. HTTP uses different requests to handle data, with the two most common being GET and POST requests. During a GET request, the client requests a specific file be sent from the server. A POST request, however, entails that the client requests permission to upload information to the server, such as content of a form. HTTP has in most cases been replaced with the more secure HTTPS-protocol that uses end-to-end encryption.

3) *RTSP*: RTSP, or Real Time Streaming Protocol, is an application layer protocol used to stream media [20].

4) *TCP*: TCP, or Transmission Control Protocol, is a transport layer protocol [21]. The main feature of TCP is that it guarantees that data arrives in full and in order. This can cause delays and higher latency because packets that get lost in transit have to be retransmitted.

5) *Telnet*: Telnet is an unencrypted network protocol that is mostly used for remote access to a computer [22]. It can also be used for other text-based applications, such as automation.

6) *UDP*: UDP, or User Datagram Protocol, is a transport layer protocol [21]. UDP is built for speed and high data throughput, and thus allows for occasional packet loss during transit, as long as the remaining packets arrive quickly.

7) *Wi-Fi / IEEE 802.11*: IEEE 802.11 or Wi-Fi, as it is commonly known, is a protocol that provides wireless network connection [23].

C. Attacks

1) *Application layer flooding*: A flooding attack is a type of denial of service attack (see section III-C4), that works by flooding a system or service with a large amount of data, to the point where it can no longer work as intended and thus preventing legitimate users from using it [24].

2) *Brute-force attack*: A brute-force attack iterates through all available combinations of a set of characters, and uses these, for example, as usernames and/or passwords during an attempted authentication [25].

3) *Cross Site Scripting (XSS)*: Cross Site Scripting is an inception attack, where an attacker attempts to input code in different input fields on a website, with the aim that the web server executes it [26]. If user input is sanitized, meaning that the input is never treated as anything but text, these types of attacks can be avoided, as potentially malicious code can never be executed.

4) *Denial of service attack (DoS)*: Denial of service attacks are a type of network based attacks, that in various ways aim to disable a service, and by doing so inhibit legitimate users from accessing it [27].

5) *Dictionary attack*: A dictionary attack uses a dictionary, for example in the form of a wordlist, that often contains common usernames or passwords [28]. During the attack, a program attempts to log in to a target website using the items in the dictionary as credentials, iterating through all combinations.

6) *Man-in-the-middle attack*: A man-in-the-middle attack is a form of interference attack, where an attacker places themselves in the middle of a communication channel [29]. During the attack, the communication between two units is relayed through an attacker, who can either simply eavesdrop or alter the content, while the communicating parties still believe they are communicating directly.

7) *Slowloris attack*: A slowloris attack [30] is a type of DoS attack (see section III-C4). The attack has low requirements on bandwidth and thus can be launched from any “ordinary” computer, as opposed to other types of DoS attacks. The attack creates a specified and large number of connections and then attempts to keep them all active by sending a small amount of data to the web server from each connection. When a legitimate user then tries to connect, the web server can not handle the request since there are already too many connections active. Not all web servers are vulnerable to a slowloris attacks, as it is primarily web servers that handle large amount of concurrent connections poorly that are affected. This means that thread-based web servers are more susceptible to slowloris attacks than event-based web servers.

D. Concepts

1) *Hashing*: Hashing is a method of irreversibly mapping a sequence of characters, for example a password, to a large, fixed length, sequence of apparently random characters [31]. Hashing is often used to store passwords, because even if the hash sequence is made public, the underlying password remains private.

2) *Insecure direct object reference*: Insecure direct object reference (IDOR) is a type of vulnerability that is caused when a web server or similar application does not authenticate credentials properly when accessing a resource directly, for example when changing a URL [32]. The implication of this malfunction is unauthorized access to information that should be protected.

3) *MD5*: MD5 or message-digest algorithm is a hashing function [33]. MD5 has previously been thought to be cryptographically secure, but has since been cracked, and is therefore not considered suitable for use within cybersecurity [34].

4) *Packet*: A packet is a segment of a message that is sent over a network. Packets contain binary data that often needs to be processed in some way [35].

IV. METHODOLOGY

A cybersecurity evaluation can be performed using different methods, with more or less insight into the product. A black box approach implies that the evaluation is performed without any assistance from the manufacturer, as opposed to a white box approach where the manufacturer is involved in the process, and can provide additional information and permissions [36].

The system under consideration was not developed by the retailer. It could, therefore, be difficult to gain access to the necessary data to perform a white box analysis, and a black box approach was consequently adopted. Using a black box approach can also make certain types of penetration test significantly more difficult and time-consuming to perform, and due to the delimitations of the project, some tests were excluded.

The methodology is further described in several sections, describing the methodology of the different phases of the project.

A. Information gathering

An important part of the initial process is to gather information about the system under consideration. Because the evaluation is conducted using a black box approach, the information gathering process becomes more extensive and time-consuming. This is because no information is used other than that which is publicly available or has been discovered during the process. To ensure that the information gathering process is sufficiently comprehensive, the process was separated into four steps [36].

1) *Using the system as intended*: To create an overview of the functionality of the system, the camera was set up following the instructions. This doubled as an investigation of possible use cases, to help determine the functionality of the camera.

2) *Source code analysis*: No source code of the Android Application was available online. However, the APK file of the application could be obtained from Google Play. This file was then processed through decompiler.com (see section III-A3), resulting in partial Java files and a few compiled library files in .so format. The .so files were further processed through the reverse engineering tool Ghidra (see section III-A4), resulting in a set of partial c-files, that were further examined.

3) *Network and traffic analysis*: The communication to and from the system was examined by capturing the transmitted data packets and analyzing them, using Wireshark (see section III-A10). Wireshark provides information about where the packets came from, where they were sent to, what protocol was used, and what payload was sent.

4) *Port scanning*: By using the tool NMAP (see section III-A6), information was acquired regarding what ports were open on the system and what services they were most probably running. This was then manually confirmed by attempting to access the services in their intended way.

B. Threat modelling

A necessary part of a cybersecurity evaluation is a comprehensive threat modelling. This section describes the methodology used in order to perform such a modelling, which can be useful both during product development and during a cybersecurity analysis of an existing device, which is the case for this project. According to Guzman and Gupta [36] the process of threat modelling an IoT device can be broken up into six steps: *identifying IoT assets, create an IoT-device architecture overview, decompose the IoT-device, identify threats, document threats, and rate the threats*. During the first step the aim is to identify all assets that could potentially be exploited in the system, and document these. The second step results in the creation of an overview of the architecture of the device, and can be further divided into three substages:

- describe and document the functionality and features of the device
- create an architectural diagram that describes the system
- identify and document the technologies used

These are all a part of the threat model, which will then be expanded upon during the subsequent steps. The third step is to decompose the IoT device, identify possible entry points into the system, and use this to expand the threat model diagram. Following the completion of the system analysis and creation of the threat model diagram are steps four, identifying threats, using STRIDE (see section IV-B1), and five, documenting these threats. The sixth and final stage of threat modelling concerns a rating of the discovered threats, using the DREAD rating system (see section IV-B2).

1) *STRIDE*: There are several ways to classify security threats. One such model is STRIDE [36], [37]. STRIDE provides six categories of threats:

- **Spoofing**: Impersonating an actor within the system
- **Tampering**: Modifying or sabotaging something in the system
- **Repudiation**: Denying doing something whether it was done, or not
- **Information disclosure**: Information being exposed to unauthorized users
- **Denial of service**: Disabling the system or service preventing legitimate users from using it
- **Elevation of privilege**: Gaining higher privilege within a system and thereby being able to execute operations that should be restricted

This categorization is useful when documenting threats, and also as a method of discovering them, by going through each category and examining the system [36], [37], which was done in this project.

2) *DREAD*: As with STRIDE there are several ways to determine the severity of a security risk. One rating system

that is frequently used is DREAD [36], which is mnemonic for the following:

- **Damage potential**: How severe the damages of a successful attack would be?
- **Reproducibility**: How easy is it to perform the attack?
- **Exploitability**: How easy is it to create a program that performs the attack?
- **Affected users**: How many users are affected?
- **Discoverability**: How easy is it to discover this vulnerability?

Each of these are given a rating ranging from one to three, and their sum determines which threat is most critical [36].

C. Threat traceability matrix

The information gathered during the threat modelling results in a number of concrete vulnerabilities and attacks that should be considered for further evaluation during penetration testing. The vulnerabilities can be summarized in a threat traceability matrix, which provides a useful overview of the system analysis that has taken place. According to [38] a threat traceability matrix contains the following information for each discovered threat, and its associated attack:

- **the attack**: what is the attack?
- **the threat agent**: who could or would carry out this attack?
- **the affected asset**: what asset is compromised during this attack?
- **the attack surface**: through what surface of the device is the attack conducted?
- **the attack goal**: what is the goal of the attack?
- **the attack impact**: what are the potential impacts of a successful attack?
- **estimated exploitability**: how difficult is the attack to perform?
- **was the attack attempted?**: has the attack been attempted?
- **results of penetration tests**: what were the results of the attacks that went on to penetration testing?

Exploitability and attack impact fall under the DREAD rating system, and is therefore not also included in a separate table for the threat traceability matrix.

D. Responsible disclosure

To prevent users of the IP camera from being exposed to the threats discovered in this cybersecurity evaluation, the vendor Biltema was given 90 days of notice to allow for threat mitigation, prior to publication [39]. The final report, and a compilation of all threats found, were sent to Biltema.

E. Penetration testing

To evaluate the vulnerabilities of the system, several penetration tests were performed. The method for each individual test is described in their respective subsection under section VII.

V. THE SYSTEM UNDER CONSIDERATION

The system under consideration is an IP- and baby camera from Biltema, seen in Fig. 1. The content of this section is a combination of the results from the initial information gathering phase (see section IV-A) and information discovered during the penetration testing.

The camera's hardware features are:

- A microphone to listen to the surrounding area.
- A speaker to playback audio.
- A light sensor, to enable the camera to automatically turn on night vision when needed.
- Several infrared light diodes for night vision.
- A MicroSD Card slot to store images and videos locally.
- A temperature sensor.
- An air humidity sensor.
- Two motors that allows the camera to turn freely, both horizontally and vertically.

A user can control and access the camera through:

- A mobile application, available for both Android and iOS.
- A locally hosted website, that uses HTTP.
- An ONVIF [40] compatible application.

The camera can communicate using:

- Wi-Fi to wirelessly connect to LAN.
- An Ethernet cable to connect to LAN.
- An unprotected Wi-Fi Hotspot to enable configuration of the system wirelessly and set it up for use on the LAN.

It was attempted to configure the camera using Wi-Fi according to the instructions in the manual, available in Appendix A. This was, however, not successful despite the camera's marketed Wi-Fi compatibility.

The camera also communicates with a cloud. The cloud is used to authenticate credentials sent from the mobile application when a user is logging in. If the authentication is successful, the application can communicate directly to the camera via an IP address supplied from the cloud.

Additionally, the camera has a Telnet server that is password protected. The credentials for the Telnet server were not supplied, therefore the Telnet server is most likely not meant to be used by the end user.

VI. THREAT MODEL AND THREAT TRACEABILITY MATRIX

This section presents the results of the threat model, resulting in a diagram (Fig. 2). Based upon this threat model, a threat traceability matrix has been developed, with regard to the delimitation set in II-C. The threat traceability matrix is divided into a risk analysis (Table I), threat analysis (Table II), and a DREAD rating (Table III). The assets of the system that could possibly be compromised during an attack are listed below:

- **Android Application:** The application available on Google Play
- **iOS Application:** The application available on the App store
- **Web server application:** web server hosted by the camera
- **Cloud:** The server hosting the authentication service



Fig. 1. A photograph of the camera

- **Hardware:** This includes the following

- Camera feed
- Speaker
- Microphone
- Air humidity sensor
- Temperature sensor

- **Firmware:** The operating system that is hosting the web server and handles all connections to and from the camera
- **Camera credentials:** Login credentials authenticated by the cloud and used in the mobile applications
- **Web server credentials:** Login credentials used on the web server

The following are explanations of important elements visualized in the threat model Fig. 2:

- **Telnet:** The two connections marked as Telnet use the Telnet protocol, which is used to open a remote terminal on the IP camera, and thereby access the firmware.
- **Custom:** The connections between the application and the camera uses a custom application layer protocol, which does not have a name, and is based on the transport layer protocol UDP.
- **After authentication:** This connection can only be used after the cloud has authenticated the credentials supplied by the user in the mobile application.

VII. PENETRATION TESTING

To test the eight threats outlined in the threat traceability matrix, ten penetration tests have been developed. The aim of each penetration test is to explore the corresponding threat by attempting to create a proof of concept of an attack.

In the following section, the methodology, results, and discussions regarding each of the ten individual penetration

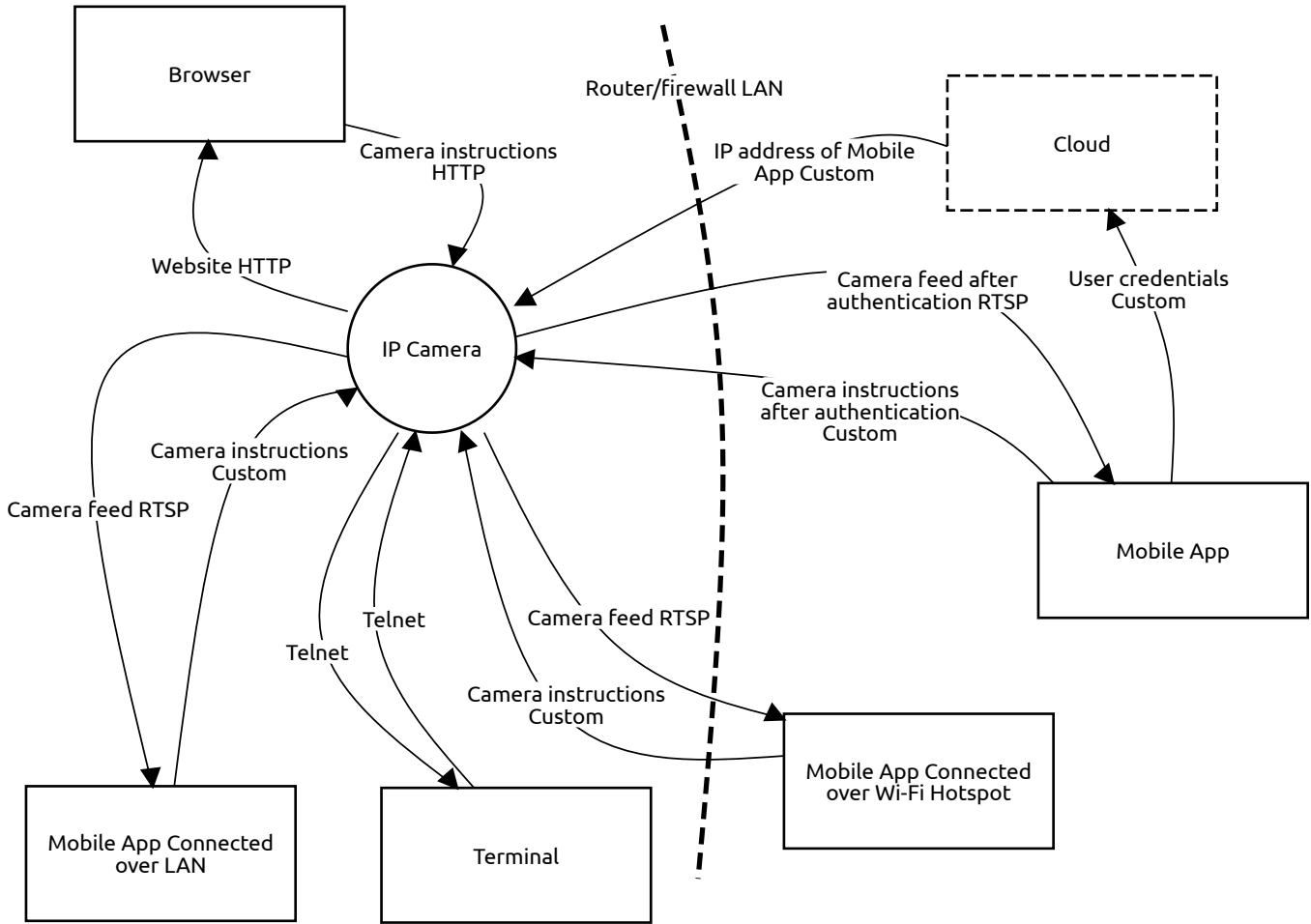


Fig. 2. Threat model made with OWASP Threat Dragon [15].

tests that have been conducted are presented. Each penetration test is connected to a distinct threat, and is therefore also represented as a row each in Table I-III, where more details regarding the relevant threat can be found.

A. Dictionary attack on Telnet

1) *Introduction:* The firmware of the system is exposed through a Telnet server. This makes the camera vulnerable to an attacker trying to guess the credentials. One systematic method of doing this is by using a dictionary attack.

2) *Method:* The application Hydra (see section III-A5) was used to perform the attack. Hydra was used to attempt to log in to the Telnet server, using two different wordlists containing common passwords [41]. Firstly, a short list of 23 commonly used IP camera passwords was used as both username and password. Secondly, a large password file containing 1310542 unique passwords was used, in combination with the usernames “root” and “user”, respectively.

3) *Results:* No correct credentials were obtained. The duration of the two tests using the longer list was 24 hours.

4) *Discussion:* Since the credentials were not found, it is hard to conclude that the product is insecure for this type of threat. However, a more experienced hacker with a better wordlist, such as described in [42], could have been more

successful. But since all very common combinations were tested, such as admin/admin and root/password, it does seem like the password was at least changed, or that less obvious credentials were chosen to begin with. This in turn indicates that the manufacturers had this threat in mind when developing the camera.

B. Brute-force attack on Telnet

1) *Introduction:* The firmware of the system is exposed through a Telnet server. This makes the camera vulnerable to an attacker trying to guess the credentials. One systematic method of doing this is by using a brute-force attack.

```
hydra -l admin -x 1:10:a1 Telnet://IP:23
hydra -l Admin -x 1:10:a1 Telnet://IP:23
hydra -l root -x 1:10:a1 Telnet://IP:23
hydra -l Root -x 1:10:a1 Telnet://IP:23
```

Fig. 3. Hydra brute-force commands

2) *Method:* The application Hydra was once again used, but this time was executed with the parameter of `-x` to perform a brute-force attack. The four commands, seen in Fig. 3, create passwords of length 1-10 using all lowercase letters

TABLE I
THREAT ANALYSIS: A PART OF THE THREAT TRACEABILITY MATRIX

#	Vulnerability	Threat	STRIDE	Attack	Attack goal
1	Having an open Telnet port on the device.	An attacker cracking the credentials of the Telnet server.	Elevation of privilege.	Dictionary attack or brute-force attack on Telnet.	Gaining root access to embedded operating system.
2	Using a web server for recording and accessing the camera feed.	An attacker disabling the web server.	Denial of service.	Slowloris or application layer flooding on web server.	Disabling the camera, thus making it unsuitable as a security camera.
3	Not disabling the configuration hotspot when the camera is connected using Ethernet.	An attacker uses the hotspot outside intended use.	Elevation of privilege.	Connecting to the Wi-Fi hotspot.	Accessing all features of the camera, such as watching the video feed and changing settings.
4	The default credentials are weak and unchangeable.	An attacker trying all possible combinations of credentials.	Spoofing, Elevation of privilege.	Brute-force attack on cloud.	Stealing the credentials to every camera of this model.
5	Insecure direct object reference (IDOR) on the local web server.	An attacker changing the camera setting without access to the camera credentials.	Elevation of Privilege, Tampering.	Accessing resources directly on web server.	Gaining full access to the cameras settings.
6	Using a weak hashing algorithm (MD5) for passwords sent over the Internet.	An attacker stealing the camera credentials over the Internet.	Information disclosure, Spoofing.	MITM attack on mobile application.	Stealing credentials and gaining unauthorized access to the system.
7	Sending unencrypted credentials between a browser and the web server.	An attacker stealing the web server credentials.	Information disclosure.	MITM attack on web server.	Stealing credentials and gaining unauthorized access to the system.
8	Not sanitizing the input from the user.	An attacker executing malicious code on the web server.	Tampering, Information disclosure, Elevation of privilege.	XSS attack on web server.	Executing code that could break the camera or expose private information.

and numbers and attempts to log in with usernames admin, Admin, root, and Root, respectively.

3) *Results:* After running the attack with a fixed username and brute-forced passwords, it was quickly evident that the attack was infeasible. The camera was at most able to process 400 login attempts per minute, and this therefore limited how many attempts could be completed within a reasonable time frame. Therefore, a decision was made to discontinue the attack. A complete attack would not be completed within our life-time.

4) *Discussion:* Due to the limitations of the system, with a maximum login handling rate of 400 requests per minute, and assuming a sufficiently complicated password, it seems unlikely that an attacker would attempt to brute-force it. If, for example, the password is eight characters long and contains a combination of numbers and letters (both uppercase and lowercase), and if the hardware is only capable of handling 400 login attempts per minute, then it would mean that there are $(26 + 26 + 10)^8$ different combinations, which would require $\frac{(26+26+10)^8}{400} \approx 5.46 \cdot 10^{11}$ minutes or over 10^6 years, which is clearly infeasible.

C. Slowloris attack on web server

1) *Introduction:* By using a denial of service attack, a hacker can disable the camera, and thus for example avoid being caught on camera. This particular attack does not exhaust the bandwidth of the attacker, and is therefore an attractive option among different types of DoS attacks.

2) *Method:* The method is well-known, and many implementations are readily available online. For this penetration test, a Python implementation of a slowloris attack was used, from [43]. The penetration test itself involves targeting the IP-address of the camera's local web server using the Python script.

3) *Results:* The penetration test was successful, as the slowloris attack not only disabled the website hosted on the web server, but also inhibited all communication to the camera. This made it impossible to access the camera through the application and website.

4) *Discussion:* It constitutes a severe risk that anyone with access to the LAN or in proximity to the local hotspot could easily disable the camera completely. The camera could also be compromised if it is directly exposed to the Internet, if, for example, a user wants to be able to access the web server remotely. Such exposure enables an attacker to perform the

TABLE II
ATTACK ANALYSIS: A PART OF THE THREAT TRACEABILITY MATRIX

#	Attack	Attack Surface	Affected Asset	Threat Agent	Attempted
1	Dictionary attack or brute-force attack on Telnet.	Telnet.	Firmware.	Unauthorized external attacker.	Yes.
2	Slowloris or application layer flooding on web server.	Web server.	Web server.	Unauthorized external attacker.	Yes.
3	Connecting to the Wi-Fi hotspot.	Wi-Fi Hotspot.	Hardware.	Unauthorized external attacker.	Yes.
4	Brute-force attack on cloud.	Cloud.	Camera credentials.	Unauthorized external attacker.	No.
5	Accessing resources directly on the web server.	Web server.	Hardware.	Unauthorized external attacker.	Yes.
6	MITM attack on mobile application.	Communication between application and the cloud.	Camera credentials.	Unauthorized external attacker.	Yes.
7	MITM attack on web server.	Communication between browser and web server.	Web server credentials.	Unauthorized external attacker.	Yes.
8	XSS attack on web server.	Web server.	Firmware, web server credentials.	Unauthorized external attacker.	Partially.

TABLE III
RISK ANALYSIS: A PART OF THE THREAT TRACEABILITY MATRIX

#	Threats	D	R	E	A	D	Risk score
1	An attacker cracking the credentials of the Telnet server.	3	3	3	3	3	15
2	An attacker disabling the web server.	2	3	3	3	3	14
3	An attacker uses the hotspot outside intended use.	3	3	3	2	3	14
4	An attacker trying all possible combinations of credentials.	3	3	3	2	3	14
5	An attacker changing the camera setting without access to the camera credentials.	3	3	3	3	2	14
6	An attacker stealing the camera credentials over the Internet.	3	2	2	3	3	13
7	An attacker stealing the web server credentials over LAN.	2	2	3	3	3	13
8	An attacker executing malicious code on the web server.	3	3	2	3	1	12

attack remotely or obscuring their identity via a virtual private network (VPN).

This would be useful for an attacker who wants to avoid detection. The results of this attack show that the camera is

unsuitable to use as a security camera. Since the attack was successful, it can be deduced that the camera implements a thread based web server, such as an Apache HTTP Server.

D. Application layer flood attack on web server

1) *Introduction:* An application layer flood attack is a type of DoS attack that aims to overload the targeted system, by flooding it with data. This could cause the targeted web server to crash, if no protections are in place.

2) *Method:* Similarly to the slowloris attack, see VII-C, the code for this type of attack is readily available across the internet. The attack was attempted using a script called PyFlood [44].

3) *Results:* The attack did not succeed, as the camera and its associated web page remained operational during the attack.

4) *Discussion:* The attack could have failed because of several reasons. For example, the web server could have built-in protection against common DoS-attacks, including application layer flooding. However, since the slowloris attack was successful, the web server seems to have an incomplete coverage against DoS attacks, and it is therefore possible that other implementations or DoS attacks would be successful.

E. Connecting to the Wi-Fi hotspot

1) *Introduction:* The local Wi-Fi hotspot is not turned off after the camera is set up using an Ethernet cable. This can be seen as a major design flaw of the system, which can then easily be used to access the camera feed and all settings without being connected to the same LAN.

2) *Background:* It is normal that an IP camera is set up using a local Wi-Fi hotspot. This allows the user to configure the camera wirelessly, using for example their smartphone and the associated app.

3) *Method:* To perform this penetration test, the camera was configured to connect to the LAN via Ethernet. After the configuration process, the active wireless hotspot was then connected to, and the available camera features were explored.

4) *Results:* When connected to the hotspot, there is full accessibility to the camera, using the application without a password. This provides usage of all the camera's functionalities which include watching the camera feed, recording both video and audio, playback audio, read the temperature and humidity, and, lastly, direct the camera with its built-in motors. All settings can also be accessed.

5) *Discussion:* The camera is intended to work while connected via Ethernet or Wi-Fi, and it is therefore possible that the hotspot is disabled if the camera is successfully connected via Wi-Fi. This is somewhat irrelevant, however, because the camera unit that was tested only worked when connected over Ethernet and thus the presence of the hotspot constitutes a significant security risk as an attacker only needs to be within reach of the hotspot to be able to completely control the camera.

F. Brute-force attack on cloud

1) *Introduction:* Analysis of the camera revealed that the default passwords only consisted of five numbers.

2) *Method:* The method will deliberately be left vague since this penetration test was not attempted, due to issues regarding legality.

The default username and password printed on the bottom of the camera were examined, and compared to the example provided in the manual, see Appendix A.

An attack towards the cloud could be performed by recording the network traffic from a login attempt, and then playing it back with other credentials, using an application such as PlayCap (see section III-A9). A script would have to be written to change the credentials sent each time.

3) *Results:* This attack was not attempted due to legal issues. But the security of the default credentials can still be evaluated. The usernames are all on the format of "TPJ" followed by 5 digits, such as "TPJ12345". The passwords are similar, only containing 5 digits, for example: "12345".

4) *Discussion:* Despite this attack not being attempted, it is a relevant threat to consider, since the default username and password are simple. This would make a brute-force attack feasible, since there are 10^5 possible usernames and 10^5 possible passwords. To brute-force the credentials of every camera of this model ever sold would be $10^5 \cdot 10^5 = 10^{10}$ combinations which is not an infeasible number over a longer period of time, depending on the capability of the cloud. With either the username or password given, it is near trivial to try 10^5 combinations. This makes the system vulnerable. This type of attack could enable an attacker to obtain all credentials to all cameras of this model, which is problematic.

G. Accessing resources directly on the web server

1) *Introduction:* This attack investigates the integrity of authentication process on the locally hosted website, where a user can change settings on the camera. The investigation was prompted by the lack of cookies on the website.

2) *Method:* The website was thoroughly examined while being logged-in, and all subpages that the web server was hosting were documented (see Table IV). The subpages were discovered by clicking different buttons on the home page and reviewing what HTTP GET requests were made to the web server, using Wireshark. To test this, it was then attempted to access all the documented subpages directly, without authentication.

3) *Results:* Out of all subpages discovered, only one subpage was not directly accessible (see Table IV). This means that the web server is vulnerable to IDOR (see section III-D2). The index page is an exception, since it is where a user logs in: it is therefore neither protected nor unprotected, in a sense. The only inaccessible page is the main home page of the camera, SystemSet. It loads the subpages when requested, but does not contain any settings in itself.

TABLE IV
ALL SUBPAGES OF THE WEB SERVER

Web address	Accessible
/cgi-bin/index.cgi	?
/cgi-bin/SystemSet.cgi	No.
/cgi-bin/DeviceMaintain.cgi	Yes.
/cgi-bin/NetConfCommon.cgi	Yes.
/cgi-bin/Wifi.cgi	Yes.
/cgi-bin/PortConf.cgi	Yes.
/cgi-bin/Ddns.cgi	Yes.
/cgi-bin/VideoCoding.cgi	Yes.
/cgi-bin/SystemStatus.cgi	Yes.
/cgi-bin/MotionDetection.cgi	Yes.
/cgi-bin/Sntp.cgi	Yes.
/cgi-bin/Ntp.cgi	Yes.
/cgi-bin/PathSave.cgi	Yes.
/cgi-bin/OnvifSet.cgi	Yes.
/cgi-bin/DeviceMaintain.cgi	Yes.

4) *Discussion:* This attack shows that there are fundamental design flaws in the web server, which exposes all camera settings and functionality. This, in combination with the weak default credentials, and, the choice of using an HTTP connection, makes the whole web server a security risk.

H. MITM attack on mobile application

1) *Introduction:* Sending credentials over the Internet can be hard to do securely. One option is to use encrypted traffic, or hashing (see section III-D1) the credentials before they are sent. This is not trivial to do correctly, and it is therefore important to evaluate exactly how this application handles credentials when they are to be transmitted.

2) *Background:* When a user wants to access the camera, they can do so by logging in to the mobile application, using credentials printed on the bottom of the camera.

3) *Method:* This penetration test consists of three stages, where the results of a stage affects the methods in the subsequent stage. Because of this, the method and result subsections of this penetration test are detailed for each stage.

4) *Method: Stage 1:* The first stage revolves around capturing data between the Android application and the cloud to attempt to extract credentials. In order to capture and analyze the network traffic to and from the Android application on a computer, rather than a mobile device, an Android x86 (see section III-A1) virtual machine running inside Oracle VM VirtualBox (see section III-A7) was used. This enables Wireshark to both capture and analyze the traffic between the Android application and the cloud. For the purpose of this penetration test, however, the choice of device should not make a significant difference, since the content of the traffic should be similar on both. The network traffic was captured during login attempts using both correct and incorrect credentials.

5) *Result: Stage 1:* When logging in using correct credentials, no clearly defined packet containing the credentials was intercepted. It remains a possibility that relevant packets were sent. However, since not all packets were easily decodable through either Wireshark or CyberChef (see section III-A2), some packets with relevant information could have been present.

When using the incorrect password “password” with the correct username “TPJ0336”, however, the packet seen in Fig. 4 was intercepted in the communication between the cloud and the app. Upon examination, it appears to be some form of error message, wherein the cloud returns the entered credentials to the application upon the failed login request. It can be noted that this packet contains the username “TPJ0336” in clear text. The sequence Psw19:68673695621059312222: indicates that the password is hashed.

```
DevFlag1:18:
userType0:6:
verNum0:2:
id16:773133$TPJ0333673:
SEQ1:43:
Psw19:68673695621059312222:
Ip15:188.151.205.1534:
Prot5:549834:
EPID1:34:
Flag1:08:
errorcod1:14:
TIME10:1645640667e
```

Fig. 4. Packet sent from the server in response to invalid credentials

6) *Method: Stage 2:* Since the password was most likely hashed (see section III-D1), an analysis of the decompiled source code (see section IV-A2) was conducted in order to determine the hash algorithm used. The code was searched using the names of common hash algorithms and keywords such as *SHA*, *Whirlpool*, *BLAKE*, *MD5*, *login* and *password*.

7) *Result: Stage 2:* From the source code, it was clear that the MD5 hash algorithm was used. MD5 produces a 32 character hexadecimal number, which does not correspond to the captured sequence. Continued examination of the code revealed that the hashed password was further modified using a custom function called `MakeMD5Int64`. The function takes the first 16 characters of a MD5 hash and then converts it

into base 10. This was the resulting hash that was sent in the intercepted packet.

8) *Method: Stage 3:* After determining the custom method of hashing used, A Python script was developed to mimic this behavior.

9) *Result: Stage 3:* The script that was developed performs the calculation automatically for a given password and has the capability of hashing a set of passwords, either using a Wordlist or brute-force, and comparing them to a given captured hash. The Python code is available in Appendix B.

10) *Discussion:* This attack was somewhat successful as credentials were discovered, but only when the server responded to the failed login request with an error message. Thus, only incorrect credentials could be discovered. This might, however, still yield useful information since it is possible that either the username or password were correct.

The contents of the error packets appears to be a form of summary of the data received by the cloud during the login request, with additional errors appended to the end. It is therefore probable that the hashed password is sent in the UDP packets preceding the cloud response, which were sent both during the correct and incorrect login attempts. Apart from some initial investigating, it was however not possible to examine these packets further due to lack of time.

The system is particularly vulnerable to brute-force attacks, since the default password distributed with the camera are purely numerical and only five digits long.

I. MITM attack on web server

1) *Introduction:* When a user logs into the local web server using a browser, to access the camera feed and camera settings, the connection between the browser and web server uses the HTTP protocol. This penetration test investigates the content of the login request sent over the LAN using HTTP, and attempts to extract login credentials within these captured packets.

2) *Background:* Although HTTP is an unencrypted protocol, there are cases when using HTTP can be secure. This can be done by encrypting the data before transmission.

3) *Method:* To investigate the content of the login packet, the traffic from the computer sending the login request was monitored using Wireshark. The data captured by Wireshark contains all communication to and from the computer, so a filter was applied to be able to view only HTTP packets. The contents of the remaining packets were examined manually and were queried using the search feature in Wireshark with keywords such as *password* and *user*.

4) *Results:* A packet was found to contain the username and password, in plain text with no encryption (see Fig. 5). These credentials can then be used to log into the web server, and thereby gain access to the camera feed and all camera settings.

```
HTML Form URL Encoded: application/x-www-form-urlencoded
.Form item: "Username" = "admin"
.Form item: "Password" = "admin"
```

Fig. 5. Data from the login HTTP POST packet, captured with Wireshark

5) *Discussion:* The attack requires access to the LAN and its network traffic that the camera is connected to. Some users might want to remotely access the camera through the web server, rather than through the mobile application. This would, however, require bypassing of the firewall in order to expose the camera to the Internet. This means that an attacker would only need access to whatever network the user is connected to, at the time of login, to be able to capture the credentials.

J. XSS attack on web server

1) *Introduction:* This type of attack can be effective when the user input is not properly sanitized (see section III-C3). It is a complicated attack to perform thoroughly, since determining what strings could cause issues heavily depends on the web server and firmware.

2) *Method:* Multiple strings of malformed commands were sent into different input fields on the local website. The pages that handle user input in some way were: the login page, the network settings pages, and the time synchronization page. The XSS attack was performed on these pages. The strings that were used for testing were sourced from “Writing Safe CGI Programs” [45].

3) *Results:* With limited testing, there was no success in exploiting this potential vulnerability.

4) *Discussion:* One condition that limited the testing was that the web server uses Common Gateway Interface (CGI) and thus it is unknown what programming language handles the user input. This makes it harder to design the payloads to send because the payload needs to be constructed based on the programming language.

VIII. RESULTS

The results of each individual penetration test can be found in their respective subsection under section VII. A summarization of the results is presented in Table V.

IX. DISCUSSION

The evaluation of the IP camera has resulted in the discovery of several vulnerabilities. Discussions connected to each penetration test can be found in their respective subsection in section VII. The demonstrates vulnerabilities pose limits to the secure usability of the camera. For example, the lack of protection against DoS attacks means that the camera is unsuitable for security applications, as it can easily be disabled. Similarly, it should not be used for situations where a compromised camera would lead to a breach in privacy. This is because an attacker in close proximity to the camera will have full access to the camera feed and all other features through the hotspot.

As can be observed in the DREAD-analysis (see Table III), many of the threats have a high discoverability rating. This, along with the number of vulnerabilities, suggest that the manufacturer has not considered security to be of prime importance in the design process, despite not using default Telnet credentials.

While several penetration tests based upon these threats have been performed, there are still many to consider, both

TABLE V
SUMMARIZED RESULTS OF EACH INDIVIDUAL PENETRATION TEST

#	Attack	Result	Successful
1a	Dictionary attack on Telnet.	No credentials were found.	No.
1b	Brute-force attack on Telnet.	No credentials were found.	No.
2a	Slowloris attack on the web server.	The slowloris attack successfully disabled the server.	Yes.
2b	Application layer flood attack on the web server.	The application layer flood attack did not disable the server.	No.
3	Connecting to the Wi-Fi hotspot.	Easy access to the camera and all its functionality.	Yes.
4	Brute-force attack on cloud.	Not attempted.	No.
5	Accessing resources directly on the web server.	The camera settings were successfully changed by exploiting an IDOR vulnerability on the web server.	Yes.
6	MITM attack on mobile application.	The credentials of a failed login attempt were captured and decrypted. However, during a correct login attempt no credentials were found.	Partially.
7	MITM attack on web server.	A HTTP POST request containing the username and password in plain text were captured.	Yes.
8	XSS attack on web server.	Due to limited time no extensive XSS attack could be performed, and the limited testing yielded no results.	No.

in regard to vulnerabilities found in this project but also vulnerabilities yet to be discovered. It is important to note that even if all confirmed vulnerabilities in this report are mitigated, there will likely still be vulnerabilities remaining.

Due to the limited time frame of the project, certain vulnerabilities could not be fully tested, as the priority was to complete the most relevant penetration tests in time. Future work could include continuing the unfinished XSS attack, further exploring the possibilities of a dictionary attack on Telnet and looking for additional vulnerabilities. It could also be relevant to attempt a white box analysis of the system (see discussion regarding white box vs. black box in section IV), and thereby be able to determine the security of the cloud.

X. CONCLUSIONS

The aim of this study was to evaluate the cybersecurity of an IP- and baby camera from Biltema. The results of the threat modelling and subsequent penetration testing

confirmed several significant vulnerabilities. Several exploits were confirmed, including methods of disabling the camera as well as gaining unauthorized access to all its features. This confirms that the camera cannot be used securely without software updates being released to mitigate the vulnerabilities discovered in this study.

APPENDIX A

BILTEMA MANUAL

APPENDIX B

MD5 PYTHON CODE

ACKNOWLEDGMENT

The authors would like to thank their supervisor Pontus Johnson for valuable input and interesting discussions.

REFERENCES

- [1] Open Web Application Security Project (OWASP). (2022, May) OWASP Top 10. [Online]. Available: <https://owasp.org/Top10/>
- [2] —. (2022, May) Internet of Things (IoT) Top 10 2018. [Online]. Available: https://wiki.owasp.org/index.php/OWASP_Internet_of_Things_Project#tab=IoT_Top_10
- [3] I. Kols and N. Hjärne, “IoT Security Assessment of a Home Security Camera,” Bsc. thesis, KTH, Stockholm, Sweden, 2020.
- [4] J. Larsson, “Are modern smart cameras vulnerable to yesterday’s vulnerabilities?” Msc. thesis, KTH, Stockholm, Sweden, 2021.
- [5] J. Fjellborg, “Identifiering och Utnyttjande av Sårbarheter hos en IP-Kamera,” Bsc. thesis, KTH, Stockholm, Sweden, 2021.
- [6] H. Georgiev and A. Mustafa, “Hacking commercial IP cameras: Home Surveillance,” Bsc. thesis, KTH, Stockholm, Sweden, 2021.
- [7] Chih-Wei Huang. (2022, Apr.) Android-x86: Run Android on your PC. [Online]. Available: <https://www.android-x86.org/>
- [8] GCHQ. (2022, Apr.) CyberChef: The Cyber Swiss Army Knife. [Online]. Available: <https://github.com/gchq/CyberChef>
- [9] (2022, Apr.) Java/APK decompiler online. [Online]. Available: <https://www.decompiler.com/>
- [10] National Security Agency. (2022, Apr.) National Security Agency/Central Security Service. [Online]. Available: <https://www.nsa.gov/>
- [11] —. (2022, Apr.) Ghidra: Software Reverse Engineering Framework. [Online]. Available: <https://github.com/NationalSecurityAgency/ghidra>
- [12] M. Heuse. (2022, Apr.) Hydra github. [Online]. Available: <https://github.com/vanhauser-thc/thc-hydra>
- [13] G. Lyon. (2022, Apr.) Nmap: the network mapper - free security scanner. [Online]. Available: <https://nmap.org/>
- [14] Oracle. (2022, Apr.) Oracle VM VirtualBox. [Online]. Available: <https://www.virtualbox.org/>
- [15] Open Web Application Security Project (OWASP). (2022, Apr.) OWASP Threat Dragon. [Online]. Available: <https://owasp.org/www-project-threat-dragon/>
- [16] A. Ott. (2022, Apr.) PlayCap. [Online]. Available: <https://github.com/signal11/PlayCap>
- [17] Wireshark Foundation. (2022, Apr.) Wireshark: Go Deep. [Online]. Available: <https://www.wireshark.org/>
- [18] Nationalencyklopedin. (2022, Apr.) Ethernet. [Online]. Available: <http://www.ne.se/uppslagsverk/encyklopedi/lng/ethernet>
- [19] —. (2022, Apr.) HTTP. [Online]. Available: <http://www.ne.se/uppslagsverk/encyklopedi/lng/http>
- [20] H. Schulzrinne, “Real Time Streaming Protocol (RTSP),” Internet Requests for Comments, RFC Editor, RFC 2326, Apr. 1998. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc1654.txt>
- [21] Internetstiftelsen. (2022, Apr.) TCP- och UDP-nivån förklarade. [Online]. Available: <https://internetstiftelsen.se/guide/introduktion-till-ip-internet-protocol/tcp-och-udp-nivan/>
- [22] Nationalencyklopedin. (2022, Apr.) Telnet. [Online]. Available: <http://www.ne.se/uppslagsverk/encyklopedi/lng/telnet>
- [23] L. Peterson and B. Davie. (2022, Apr.) Computer networks: A systems approach: 2.7.2 wi-fi (802.11). [Online]. Available: <https://book.systemsapproach.org/direct/wireless.html#wi-fi-802-11>
- [24] Cloudflare. (2022, Apr.) HTTP flood attack. [Online]. Available: <https://www.cloudflare.com/learning/ddos/http-flood-ddos-attack/>
- [25] —. (2022, Apr.) What is a brute force attack? [Online]. Available: <https://www.cloudflare.com/learning/bots/brute-force-attack/>
- [26] —. (2022, Apr.) What is cross-site scripting? [Online]. Available: <https://www.cloudflare.com/learning/security/threats/cross-site-scripting/>
- [27] —. (2022, Apr.) What is a denial-of-service (DoS) attack? [Online]. Available: <https://www.cloudflare.com/learning/ddos/glossary/denial-of-service/>
- [28] Kaspersky. (2022, Apr.) Dictionary attack. [Online]. Available: <https://encyclopedia.kaspersky.com/glossary/dictionary-attack/>
- [29] NIST Computer Security Resource Center. (2022, Apr.) man-in-the-middle attack (MitM). [Online]. Available: https://csrc.nist.gov/glossary/term/man_in_the_middle_attack
- [30] Cloudflare. (2022, Apr.) Slowloris DDoS attack. [Online]. Available: <https://www.cloudflare.com/learning/ddos/ddos-attack-tools/slowloris/>
- [31] NIST Computer Security Resource Center. (2022, May) hash. [Online]. Available: <https://csrc.nist.gov/glossary/term/hash>
- [32] Open Web Application Security Project (OWASP). (2022, Apr.) Insecure Direct Object Reference Prevention Cheat Sheet. [Online]. Available: https://cheatsheetseries.owasp.org/cheatsheets/Insecure_Direct_Object_Reference_Prevention_Cheat_Sheet.html
- [33] D. Liu, “Chapter 3 - an introduction to cryptography,” in *Next Generation SSH2 Implementation*, D. Liu, M. Caceres, T. Robichaux, D. V. Forte, E. S. Seagren, D. L. Ganger, B. Smith, W. Jayawickrama, C. Stokes, and J. Kancirz, Eds. Burlington: Syngress, 2009, pp. 41–64.
- [34] M. Mao, S. Chen, and J. Xu, “Construction of the Initial Structure for Preimage Attack of MD5,” in *2009 International Conference on Computational Intelligence and Security*, vol. 1, 2009, pp. 442–445.
- [35] Cloudflare. (2022, Apr.) What is a packet? [Online]. Available: <https://www.cloudflare.com/learning/network-layer/what-is-a-packet/>
- [36] A. Guzman and A. Gupta, *IoT Penetration Testing Cookbook: Identify vulnerabilities and secure your smart devices*. Birmingham, UK: Packt Publishing, 2017.
- [37] A. Shostack, *Threat modeling: designing for security*, 1st ed. Indianapolis, IN: John Wiley & Sons, 2014.
- [38] Division of Network and Systems Engineering — KTH. (2022, Apr.) Threat Traceability Matrix. [Online]. Available: https://nse.digital/pages/thesis_guidelines/threat_traceability_matrix.html
- [39] —. (2022, Apr.) Responsible disclosure. [Online]. Available: https://nse.digital/pages/thesis_guidelines/responsible_disclosure.html
- [40] ONVIF. (2021, Dec.) ONVIF Streaming Specification. [Online]. Available: <https://www.onvif.org/specs/stream/ONVIF-Streaming-Spec.pdf>
- [41] J. Lee. (2022, Apr.) Wordlists. [Online]. Available: <https://github.com/jeanphorn/wordlist>
- [42] J. Li, E. Zeigler, T. Holland, D. Papamichail, D. Greco, J. Grabentein, and D. Liang, “Common passwords and common words in passwords,” in *Trends and Innovations in Information Systems and Technologies*. Cham, Switzerland: Springer International Publishing, 2020, pp. 818–827.
- [43] G. Yaltirakli. (2015) Slowloris. [Online]. Available: <https://github.com/gkbrk/slowloris>
- [44] D4Vinci. (2021, Jun.) Pyflood. [Online]. Available: <https://github.com/D4Vinci/PyFlood>
- [45] T. A. Fine. (2008) Writing Safe CGI Programs. [Online]. Available: <https://hea-www.harvard.edu/~fine/Tech/cgi-safe.html>

Integrating the Meta Attack Language in the Cybersecurity Ecosystem: Creating new Security Tools Using Attack Simulation Results

Björn Thiberg and Frida Grönberg

Abstract—Cyber threat modeling and attack simulations are new methods to assess and analyze the cybersecurity of IT environments. The Meta Attack Language (MAL) was created to formalize the underlying attack logic of such simulations by providing a framework to create domain specific languages (DSLs). DSLs can be used in conjunction with modeling software to simulate cyber attacks. The goal of this project was to examine how MAL can be integrated in a wider cybersecurity context by directly combining attack simulation results with other tools in the cybersecurity ecosystem. The result was a proof of concept where a small DSL is created for Amazon EC2. Information is gathered about a certain EC2 instance and used to create a model and run an attack simulation. The resulting attack path was used to perform an offensive measure in Pacu, an AWS exploitation framework. The result was examined to arrive at conclusions about the proof of concept itself and about integrating MAL in the cybersecurity ecosystem in a more general sense. It was found that while the project was successful in showing that integrating MAL results in such manner is possible, the CAD modeling process is not an optimal route and that other domains than the cloud environment could be targeted.

Sammanfattning—Cyberhotmodellering och attacksimuleringar är nya metoder för att bedöma och analysera cybersäkerheten i en IT-miljö. Meta Attack Language (MAL) skapades för att formalisera den underliggande attacklogiken för sådana simuleringar genom att tillhandahålla ett ramverk för att skapa domain-specific languages (DSL). En DSL kan användas tillsammans med modelleringsprogramvara för att simulera cyberattacker. Målet med detta projekt var att undersöka hur MAL kan integreras i ett bredare sammanhang genom att direkt kombinera MAL-resultat med andra verktyg inom IT-säkerhet. Resultatet blev ett koncepttest där en mindre DSL skapades för Amazon EC2. Information samlades in om en viss EC2-instans och användes för att skapa en modell och genomföra en attacksimulering. Den resulterande attackvägen användes för att utföra en offensiv åtgärd i Pacu, ett ramverk för AWS-exploatering. Resultatet undersöktes för att nå slutsatser om konceptet i sig och om att integrera MAL i IT-säkerhetens ekosystem i allmänhet. Det visade sig att även om projektet lyckades visa att det är möjligt att integrera MAL-resultat på ett sådant sätt, är CAD-modelleringsprocessen inte en optimal metodik och lämpar sig illa för syftet. Det visade sig också att andra domäner än molnmiljön skulle vara en givande inriktning.

Index Terms—Meta Attack Language, Attack Simulation, Amazon EC2, Cybersecurity.

Supervisors:

Robert Lagerström, Viktor Engström

TRITA number: TRITA-EECS-EX-2022:166

I. INTRODUCTION

Large scale IT systems and the necessity to keep them secure is an increasingly prevalent part of modern society and infrastructure. Cybersecurity is a process that often requires deep knowledge, experience and work ethic to perform adequately. A small gap in protocols, configuration or management can be difficult to detect while having catastrophic consequences. One method to gauge, analyze and improve the security of IT systems are attack simulations. Simulating the behavior and attack path of an adversary is a way to systematically tackle the issue of cybersecurity, alleviating the pressure on individual experts or analysts. What attack simulations actually entail can vary. One form of such a simulation uses attack graphs, an abstract model of an IT system using nodes and vertices to represent everything from security policies and firewalls to individual users and the relationship between them.

To provide a more formalized method of generating these attack graphs, Johnson et al. presented the Meta Attack Language (MAL) [1]. MAL is a framework for creating domain-specific languages (DSLs) that are used to model a certain domain and allow for easier generation and computation of attack graphs. The creation of DSLs allows for re-usability of the logic and structure of the domain, reducing the work needed to model and analyze a specific instance of it. MAL thus allows a separation of competencies; a service provider creates a DSL for their service while a user utilizes it to model their own specific instance, assessing its security flaws and strengths without the necessity to understand the underlying modeling logic. These domains can vary in scope and generality from a generic IT system to a very specific domain, such as a certain cloud service. Previous uses of MAL and created DSLs are outlined in the Related work section.

A. Problem formulation

Earlier contributions in the realm of MAL focus mainly on creating new DSLs, secondarily on extending the use cases of and combining existing DSLs. These DSLs are often made for the enterprise level, for large scale and exhaustive security assessments; there is a lack of attempts to integrate MAL in the larger cybersecurity ecosystem. The global skill gap in cybersecurity [2] creates a need for simple, packaged security tools that allow system administrators and other non-security professionals or laymen to perform defensive security assessments of their systems. Where commercial enterprise-level DSLs provide a solution for large domains, smaller and

more focused tools based on MAL can appeal to another group of users and use-cases.

There is also an abstract question that a new tool incorporating MAL results could help answer. MAL is developed in an academic context as a relatively self-contained project and all-encompassing solution. Cybersecurity professionals rarely depend on one single piece of software, instead using a toolkit of well-tested and established tools with specific scopes. By creating a new bundle that combines MAL with established security tools, there is an opportunity of bridging the gap between these two worlds. While established MAL DSLs are tested, this is done to validate and evaluate the correctness and completeness of the DSL itself, not for the reasons stated above.

B. Project goal and scope

The goal of this project is to examine how MAL can be integrated in a wider cybersecurity context by creating a security tool that uses attack simulation results from MAL combined with other tools in the cybersecurity ecosystem. The intended result is examining if such a bundle of tools is a feasible idea and how it could be designed. The result and design process is examined to reach conclusions about how MAL can be used in conjunction with other tools and use-cases in a general sense. This project is limited to a proof of concept, attempting to show that it is possible to use MAL results together with other tools and to use these attack simulations in a more operative manner than previously. The aim is not to create a fully-fledged security tool.

C. Report structure

The remainder of this paper is organized as follows. Section 2 describes technical background needed to understand the methodology and result. Section 3 outlines the related and previous work in MAL. Section 4 describes the methodology used to design, construct and test the proof of concept. Section 5 outlines the result of the project, including the structure of the created tool. Section 6 evaluates the project results. Section 7 discusses the project, its value compared to other ideas and choice of components. Finally, section 8 arrives at a conclusion and outlines specific potential future work.

II. TECHNICAL BACKGROUND

A. Attack simulations

Attack simulation is a widely used concept and does not have an agreed upon definition, since the concept is both recent and abstract in nature. In this paper an *attack simulation* is defined as a Monte-Carlo simulation that samples the probability distributions of a given attack graph and then by using a shortest path algorithm between assets the most probable attack paths are returned. This is the manner in which the software SecuriCAD, used in this project, handles and defines attack simulations [3].

B. MAL structure and logic

At the core of MAL is a formalism and structure to create DSLs. According to this formalism, vertices in a graph represent objects, also called assets. Each object is a part of a class that has a set of attack steps associated with it. Classes can also have an associated set of defences.

The attack steps are represented by directed edges with an associated weight, representing the time it takes to perform the attack step. Attack steps can be separate from each other but lead to the same result, such as one step utilizing password access and another bypassing firewall rules (OR), both leading to the same endpoint from different entry points. Attack steps can also consist of a combination of separate steps, with all of them being requirements to reach an asset (AND).

Defences has a state of either TRUE or FALSE and can act as parents to attack steps, such that an attack step can only be preformed if the defense is FALSE. Defences can also affect the time it takes to perform a certain attack step. [1]

C. SecuriCAD

SecuriCAD is a CAD tool for modeling and assessing the security of IT-environments. It was first presented by Ekstedt et al. in [4] and was developed by Foreseeti, a company spun out of KTH research. SecuriCAD can be run in conjunction with MAL, combining to create a fully-fledged security assessment based on attack simulations. It uses MAL-created DSLs to create an attack graph that is then used to analyze the environment and is responsible for the actual simulations that are then used to analyze the probability of an attacker reaching or compromising assets. [5]

D. Amazon Web Services (AWS)

Amazon Web Services is the largest cloud infrastructure service by market share, offering more than 200 different cloud services. [6] [7] Amazon Elastic Compute Cloud (EC2) is a part of AWS as one of its services, allowing users and companies to run and manage cloud-based virtual machines (VMs) [8], capable of running a multitude of operating systems, such as Linux distributions or Windows Server [9]. Access to Amazon EC2 instances are managed by public-key cryptography. A public key is stored and connected to a specific EC2 instance, AWS user or a created user with specific permissions, with a connected private key. These form a set of credentials validating the user and granting a certain amount of access and privileges is used to manage the EC2 instance using the AWS dashboard [10]. Every EC2 instance is also connected to a certain security group. A security group is a set of rules governing inbound and outbound traffic. These rules can, for example, set a certain range of IP addresses from which remote access to the VM is allowed.

III. RELATED WORK

Since the creation and presentation of MAL in [1], several DSLs have been created and used to model instances of IT systems and infrastructure. Katsikeas et al. have presented

coreLang to model a generic type IT system [11]. enterpriseLang presented by Xiong et al. provides a way to model a generic cloud system and is based on attack and defense knowledge in MITRE Enterprise ATT&CK Matrix [12].

The DSL for AWS [13] and awsLang [14] are DSLs that attempt to model the Amazon Web Services (AWS) domain. Jefford-Baker presents ALCOL which builds on awsLang to more accurately model and analyze Elastic Container Services(ECS) provided by AWS [15]. Hawasli presented a DSL that model the Microsoft Azure domain, another cloud computing service [16]. Other domains such as connected vehicles [17] [18] and electrical and power systems [19] have also been modeled. Almgren and Holm Åström uses MAL to connect the DSL:s presented in [18] and [14] to model a AWS-connected vehicle [20].

The work of Hacks et al. extends MAL by developing a method to use ArchiMate notation, a commonly used modeling tool, to create MAL instances to analyze [19]. KEBande et al. uses MAL in a machine learning context [21]. Evensjö extends enterpriseLang by providing probability distributions to possible attack paths, as well as an analysis of the financial impact of an attack and if available mitigation measures are profitable [22].

IV. METHODOLOGY

The design process of the proof of concept is based on the process model motivated, presented and demonstrated by Peffers et al in [23], dividing the six process steps in Peffers into three stages in the manner outlined below. Henceforth, the complete proof of concept itself, referring to the bundle of tools and surrounding software created for this project and any additional handlers and scripts used is combined referred to as *the artifact* in the manner used in Peffers.

1) Problem identification

A general problem identification and motivation is dealt with in the introduction of this paper. The more specific problem in the context of this section is the question of *how can an artifact combining MAL and SecuriCAD with other, established cybersecurity or IT tools be created?* Secondary problems are *how to ensure such an artifact and the process of its creation has general applicability and how such an artifact might lead to substantial and more complete tools and use-cases in future work.*

2) Solution objectives, design and development Choosing an IT environment to use as target and platform is the first objective. To ensure the generality outlined in the goal of this project, it is vital that this environment is both widely used in the present and that it will be in the future. Choosing a domain that is both accessible and shares characteristics with other domains maximizes the applicability of the results.

Since attack simulations using MAL and SecuriCAD both require an *input* in the form of a DSL and information about a specific instance, as well as produce an *output* in the form of simulation results, it is natural to place MAL/SecuriCAD at the center of a three-step process. The artifact will thus first require the use of a tool that has the capacity to gather and convey information about the targeted domain, used to model

the specific instance of the chose environment. It will also require the creation of a DSL of the targeted domain, compiled for use in SecuriCAD. Lastly, the result of the SecuriCAD modeling and simulation will in turn serve as input to a security testing tool suited for the chosen domain and capable to act on the simulation result, performing some offensive or defensive measure.

A final objective surrounding the process outlined above is the creation of software or a script that combines and directs the different inputs and outputs together, here referred to as the handler. A high-level, general-purpose programming language with compatibility and publicly available libraries and resources is well-suited for this task, since performance is of secondary importance. Putting these objectives together produces five different objectives of a solution, Fig. 1 shows the artifact on a conceptual level.

- 1) Choose and construct an instance of a widely used IT environment and create a MAL DSL for this domain.
- 2) Utilize a suitable tool connected to this environment to gather and convey information about the instance.
- 3) Use SecuriCAD to model the specific instance and perform an attack simulation.
- 4) Utilize a cybersecurity tool to perform an offensive or defensive measure against the instance based on the attack simulation results.
- 5) Create the handler conjoining the different tools into the complete artifact.

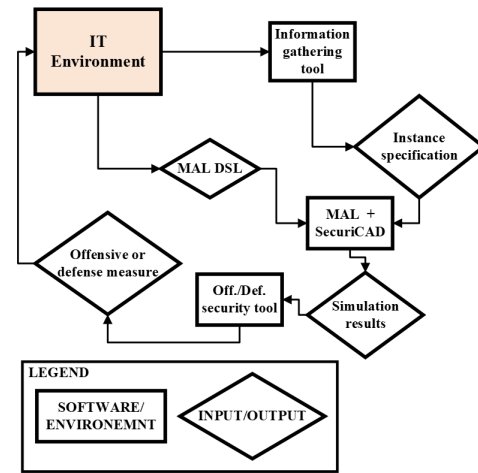


Fig. 1. Schematic artifact representation.

3) Demonstration and evaluation The artifact is demonstrated and communicated in this paper, running the handler and registering both the effect of individual steps and the end result on the targeted environment. It is evaluated based on fulfillment of project goals, including evaluating whether the artifact actually integrates MAL results with other tools in a meaningful manner and whether it allows later conclusions about integrating MAL in a wider context.

V. RESULT

The resulting artifact was a partially automated bundle of software with a purpose-built Python script handling input and

output, with MAL and SecuriCAD at its center. An Amazon EC2 Instance was setup with specific rule-sets for allowing and disallowing remote access, and a simple DSL was created to allow modeling of this specific EC2 environment. A security key allowing partial access to this AWS account was then provided and information gathered through the AWS Command-line Interface (CLI) was used to gather information and allow the manual modeling of the instance and its security rules in SecuriCAD. The EC2 instance and the AWS access key were represented as assets in SecuriCAD and the rules for only allowing specific IP addresses in inbound remote access traffic was implemented as a defense. The simulation results were then supplied as ground for executing an attack module in a modified version of Pacu, an open-source AWS exploitation framework built by Rhino Security Labs [24]. The chosen module changes the rules of the security group rules governing inbound traffic, representing an offensive measure. Fig. 2 shows the project result schematically. Following is a detailed breakdown of the artifact, corresponding to the five different objectives of a solution outlined in the methodology section above.

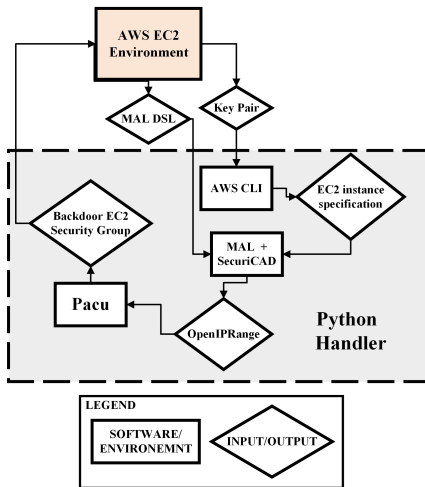


Fig. 2. Schematic result representation.

A. AWS EC2 Instance and KEXLang DSL

The chosen IT environment was Amazon Web Services, specifically the EC2 environment. Because of the confusing language and definition of the word *instance*, referring both to an EC2 virtual machine and MAL instances, the EC2 instance will henceforth be referred to as the *VM*. The choice of AWS as environment was motivated by its position as the largest of the commercial cloud services, and for the practical reasons through the fact that AWS offers hosting free of charge. Cloud hosting services are both increasingly popular and share similarities with one-another, meaning the choice of a cloud environment for this project is well motivated in terms of integrating MAL in a larger IT and cybersecurity ecosystem, achieving the objective of making the result as useful and general as possible. AWS offers a multitude of services, with EC2 being just one of them and not the only

LISTING I
KEXLANG DSL CODE EXCERPT.

```

asset EC2Instance {
  | getPrivateKey
  -> address_range.openIPRange
  | backdoorAccess
  -> access
  # restrictedIPRange
  -> access
  & access
}

| = attack step (OR)
& = attack step (AND)
-> = leads to/enables
# = defense
  
```

possible choice. However, since virtual machines are not a phenomenon specific to cloud environments, and essentially act as computers on a traditional local network in terms of security rules, and access to non-cloud tools, EC2 is well suited as a ground for this project. Note that the free usage tier used in this project is limited only in processing and storage capacity and scalability, and is "not limited to specific use cases" [25] and thus does not reduce applicability for larger EC2 environments. An AWS user and EC2 VM was created as the targeted environment for this project. The VM was a machine running the Amazon Linux Operating System, setup with standard access rules except for limiting remote access traffic to a specific IP range.

A DSL for the Amazon EC2 environment was then created, aptly named KEXLang. While DSLs that can model EC2 environments and the larger AWS domain already exist, it was decided that the benefit of using a DSL that only models the, in this case, necessary assets outweighs the value of a more extensive modeling of the instance since this artifact only entails a proof of concept. A simple DSL was thus created for the EC2 domain and compiled through the SecuriCAD back-end to provide SecuriCAD-compatible MAL logic for later modeling of the EC2 environment.

The asset representing the EC2 VM has the attack steps `getPrivateKey`, that enables the step `openIPRange`, and `backdoorAccess` as well as associations to the assets representing the IP-range and private key. Both attack steps are of the type OR and the defense `restrictedIPRange` was also implemented as well as the AND attack step `access`. Meaning that an attacker first needs to access the private key and can then open the range of allowed IP-addresses, which in turn allows backdooring the EC2 VM and accessing it. However, if the defense `restrictedIPRange` was set to FALSE then an attacker could simply access the instance without needing to access the private key and backdoor the instance. An excerpt from the DSL can be seen in Listing 1. For the entire MAL-specification, see Appendix.

B. AWS CLI information gathering

The AWS CLI was used to gather information and convey it in a readable format to be used as basis for the SecuriCAD modeling process. Specifically, the AWS CLI commands `ec2 describe_instances` and

`ec2 describe_security_groups`, returning information about all currently running EC2 VMs and their corresponding security groups respectively, in .json format to allow human readability for the modeling process. Depending on the access levels and privileges associated with the supplied AWS access key, the AWS CLI is a complete interface for interacting, examining and describing all running EC2 VMs. There are other tools that in different ways emulate AWS CLI and allow this type of information gathering and processing, such as securiCAD AWS Collector [26] or the AWS Dashboard graphical user interface. However, since AWS CLI is the native tool for interacting with AWS services and since it lends itself well to be handled by other processes it was considered as a natural choice for this project.

C. SecuriCAD modeling and simulation

Through the DSL compiled for use with SecuriCAD and with the information gathered through AWS CLI, the EC2 environment created for the artifact was manually modelled in the SecuriCAD interface. The model itself can be seen in Fig. 3. The star represents the asset with an attack step that has a consequence value over zero, which here refers to the `OpenIPRange` attack step, modifying the associating EC2 security group by opening the range of IP-addresses allowed for remote access, which in this case was set to have a consequence value of 1.

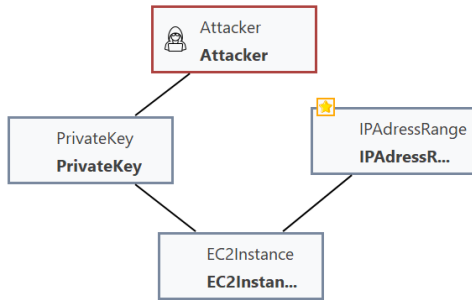


Fig. 3. SecuriCAD model. Shows assets and their current associations.

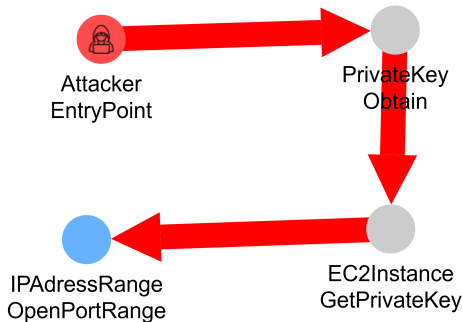


Fig. 4. Attack graph (simulation results)

The simulation results are then exported as a .csv file containing possible attack paths and the related consequence value. In this artifact, the SecuriCAD model is constructed in such a manner as to *always* return the attack step `OpenIPRange` as the result with the highest consequence, despite the probabilistic nature of SecuriCAD modeling. The resulting attack path is shown in Fig. 4.

D. Offensive measure in Pacu

Pacu is a modular framework for performing security assessments of the AWS environment [24]. It is comprised of a set of modules designed to perform some offensive measure targeting an AWS account or instance in several areas, ranging from reconnaissance to privilege escalation. Such a module is in itself a Python script and requires a set of AWS credentials (access key) and performs some offensive measure against the targeted system. Since cloud security is a young field there is a smaller array of tools available compared to those for security assessments of more traditional environments, which consists mainly of local networks. Stratus Red Team [27] was first considered in the artifact design process instead of Pacu and has, at a glance, a similar objective and scope to that of Pacu. However, Stratus differs in philosophy in that it *emulates* offensive attack techniques through creating its own EC2 VM, separate from any existing system. Pacu is well-suited for this artifact since it affects the modeled instance directly. To allow for the required Pacu module to later be run directly from the Python handler, the source code was modified. Pacu requires the creation of so called *sessions* and the import of AWS access keys before executing a module, and neither the creation of sessions or the import of keys can be run from a command-line environment. The source code was modified to automatically create a new session and import the system's default AWS access keys. Since the SecuriCAD simulation returns `OpenIPRange` as the attack step with the highest consequence set, the Pacu module `ec2__backdoor_ec2_sec_groups` is triggered, targeting the security group associated to the EC2 VM. As the name suggests, this module affects an EC2 security group. It modifies the rules governing which port and/or IP ranges are allowed for inbound traffic for different kinds of remote access, including Secure Shell (SSH). The module was run, setting the allowed range of IP addresses to `0.0.0.0/0`, which is short-hand for every possible address (in the IPv4 version). The result of the Pacu module is evaluated through manually controlling the affected security group, ensuring the rules for inbound traffic are actually modified.

E. Python handler

Resolving the final objective of the artifact design process, a Python handler script was created to bridge the different components described above. It was built to perform four different tasks to handle input and output between the components of the artifact. The script both formats data into a format readable for the next step of the artifact, and runs the processes outlined above as Python subprocesses. Firstly, the handler was designed to set a provided AWS access key-pair as the default system key for AWS. This key is then utilized in the later stages, both AWS CLI and Pacu solely use this key to access the AWS environment. Secondly, the handler triggers the AWS CLI commands and formats the instance and security group data output to give meaningful information for the SecuriCAD modeling process. It then waits for the SecuriCAD simulation result, as this CAD process is not possible to automate. Thirdly, it formats the simulation .csv output and

LISTING II
PYTHON HANDLER TERMINAL OUTPUT

```
aws_access_key_id:
> [AWS access key ID]
aws_secret_access_key:
> [AWS secret access key ID]
Instance data gathered and saved to .json
Compiling MAL DSL to SecuriCAD .jar ...
SecuriCAD .jar file now in ./
SecuriCAD output as result.csv? [y/n]
> y
OpenIPRange has consequence: 1
Running [ec2__backdoor_ec2_sec_groups]...
MODULE SUMMARY:
  1 security group(s)
  successfully backdoored.
```

finds the attack step with the highest consequence set. Finally, it translates the simulation result to its corresponding Pacu module and triggers it. A section of the terminal output is highlighted in Listing 2.

VI. EVALUATION

While the artifact was a success in terms of combining different tools and attack simulations based on MAL, the scope was limited. However, since the three steps of the artifact are clearly defined: AWS CLI, MAL/SecuriCAD and a Pacu measure based on output, utilizing the same artifact structure for future projects is possible. For example, using a more extensive DSL that contains additional assets and associations might allow a more realistic and meaningful model of the EC2 domain, and thus allow a larger amount of potential simulation results as well as their corresponding Pacu modules. However, while extending the proof of concept with the chosen components could be meaningful, an additional question is whether the methodology itself and the chosen domain, components and processes are the best way forward. Using the CAD modeling process at the core of the artifact appeared to be ill-suited for its purpose. There is no way to get around the process of modeling the targeted domain manually, heavily restricting the automation potential of a future tool and thus its usefulness. This is true whether the DSL and information gathering processes are improved or not. A shift in methodology would be finding a way to utilize the underlying modeling and simulation logic in MAL and SecuriCAD while automating or at least heavily simplifying the CAD process. As it stands, using other tools as input and output for SecuriCAD modeling is essentially just a way to automate the information gathering and security evaluation process that are already part of how MAL is meant to be used, not in any way a new methodology of performing security assessments. Additionally, while Pacu served its purpose in providing a modular and accessible interface for acting on the attack simulation results, it required modification to be used in a command-line environment and is very much a work in progress, limiting its usefulness.

VII. DISCUSSION

There are definitely arguments to be made about creating another artifact with other components. In addition to replacing

or modifying the role of SecuriCAD, a domain other than the cloud domain could be chosen. Cloud security is a young field and there are therefore fewer tools with native cloud support, and less documentation and academic work in the area. If the goal is to incorporate MAL in the security ecosystem, a well-established domain such as traditional network security could be chosen instead. Taking the step from how MAL has been used previously while simultaneously entering the field of cloud security is perhaps two steps that should not be taken at the same time. Additionally, whether targeting the cloud domain or not, basing the project on an already existing and well-tested DSL for the chosen domain could both make the project more robust and make the step from the established MAL development process smaller. This could also help solve a more technical problem; when choosing from a smaller range of potential tools there are restrictions on what OS and platforms can be used, further complicating the process of merging different tools.

In a broader sense, the question remains on whether MAL *should* be used and developed in the manner demonstrated in this project. While it is relatively easy to argue that new types of cybersecurity tools are necessary due to the current state of the industry, it is not necessarily evident where the focus should be. Other methods and frameworks for developing new forms of tools, for example by using Machine Learning/Artificial Intelligence to predict attack paths as well as critical assets are alternatives to attack simulations and MAL logic. One manner of thinking is simply that the issue of security in modern IT infrastructure is so critical that attempts to find new paths forward should be an effort on every possible front. Such a motivation notwithstanding, attack simulations and threat modeling could serve a different purpose than other tools. Machine Learning algorithms are by definition trained on past happenings, events and data points. Combining attack simulations and MAL logic with the experience and knowledge of the engineer doing the modelling could allow for a more proactive and creative methodology, as opposed to relying on neural networks and Machine Learning algorithms to perform the work for them.

VIII. CONCLUSION

In conclusion, while directly integrating MAL and attack simulations with other tools is feasible and could allow for new tools and methods for cybersecurity professionals, there are both technical and conceptual problems that make this process difficult. Choosing an established, well-tested MAL DSL as basis, and targeting an already well-examined IT security domain and using tools made for it could be the future of integrating MAL in the wider cybersecurity domain.

A. Future work

Future work includes creating another artifact with similar components; including SecuriCAD, AWS and Pacu. Every step of the process could be expanded upon to create a more extensive and improved new version of the artifact created for this project. Another future project would be to move away from CAD, replacing the SecuriCAD manual modelling

process with tools that can be automated, creating simulation based on MAL logic directly through the use of automatic information gathering tools.

APPENDIX

The DSL, the modified Pacu source code and the output from certain steps in the project can be found in this GitHub repository: gits-15.sys.kth.se/bthiberg/KEX_M6_appendix

ACKNOWLEDGMENT

The authors would like to extend their gratitude to our supervisors Viktor Engström and Robert Lagerström for their support and guidance in this project.

REFERENCES

- [1] P. Johnson, R. Lagerström, and M. Ekstedt, "A Meta Language for Threat Modeling and Attack Simulations," ser. ARES 2018. New York, NY, USA: Association for Computing Machinery, 2018. [Online]. Available: <https://doi.org/10.1145/3230833.3232799>
- [2] B. L. Jon Oltsik, "The life and times of cybersecurity professionals 2021," *information systems security association*, vol. 5, 2021.
- [3] (2022, Mar) securiCAD documentation - FAQ. [Online]. Available: <https://docs.foreseeti.com/docs/faq-2>
- [4] M. Ekstedt, P. Johnson, R. Lagerström, D. Gorton, J. Nydrén, and K. Shahzad, "Securi CAD by foreseeeti: A CAD Tool for Enterprise Cyber Security Management," in *2015 IEEE 19th International Enterprise Distributed Object Computing Workshop*, 2015, pp. 152–155.
- [5] M. Ekstedt. (2022, Mar) Foreseeeti blogpost: the Meta Attack Language (MAL). [Online]. Available: <https://foreseeti.com/meta-attack-language/>
- [6] (2021, Oct) Amazon, Microsoft & Google Grab the Big Numbers – But Rest of Cloud Market Still Grows by 27%. [Online]. Available: <https://www.srgresearch.com/articles/amazon-microsoft-google-grab-the-big-numbers-but-rest-of-cloud-market-still-grows-by-27>
- [7] (2022, Mar) What is AWS. [Online]. Available: <https://aws.amazon.com/what-is-aws/>
- [8] (2022, Mar) Amazon EC2. [Online]. Available: <https://aws.amazon.com/ec2/>
- [9] (2022, Mar) Supported operating systems. [Online]. Available: <https://docs.aws.amazon.com/systems-manager/latest/userguide/prereqs-operating-systems>
- [10] (2022, Mar) Amazon EC2 key pairs and Linux instances. [Online]. Available: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs>
- [11] S. Katsikeas, S. Hacks, P. Johnson, M. Ekstedt, R. Lagerström, J. Jacobsson, M. Wällstedt, and P. Eliasson, "An Attack Simulation Language for the IT Domain," in *Graphical Models for Security: 7th International Workshop, GraMSec 2020, Boston, MA, USA, June 22, 2020, Revised Selected Papers*. Berlin, Heidelberg: Springer-Verlag, 2020, p. 67–86. [Online]. Available: https://doi.org/10.1007/978-3-030-62230-5_4
- [12] W. Xiong, E. Legrand, O. Åberg, and R. Lagerström, "Cyber security threat modeling based on the MITRE ENterprise ATT&CK Matrix," *Softw Syst Model*, p. 157–177, Jun 2022.
- [13] V. Engström, P. Johnson, R. Lagerström, E. Ringdahl, and M. Wällstedt, "Automated Security Assessments of Amazon Web Service Environments," *ACM Transactions on Privacy and Security*, no. 1, Jan 2018.
- [14] A. Singh Viridi, "AWSLang: Probabilistic Threat Modelling of the Amazon Web Services environment," M.Sc thesis, KTH, Stockholm, Sweden, 2018.
- [15] J. Jefford-Baker, "ALCOL: Probabilistic Threat Modelling of the Amazon Elastic Container Service Domain," M.Sc thesis, KTH, Stockholm, Sweden, 2019.
- [16] A. Hawasli, "azureLang: a probabilistic modeling and simulation language for cyber attacks in Microsoft Azure cloud infrastructure," M.Sc thesis, Stockholm, Sweden, 2018.
- [17] A. Girmay Mesele, "AUTOSARLang: Threat Modeling and Attack Simulation for Vehicle Cybersecurity," 2018, M.Sc thesis, KTH.
- [18] S. Katsikeas, "vehicleLang: a probabilistic modeling and simulation language for vehicular cyber attacks," M.Sc thesis, KTH, Stockholm, Sweden, 2018.
- [19] S. Hacks, A. Hacks, S. Katsikeas, B. Klaer, and R. Lagerström, "Creating Meta Attack Language Instances using ArchiMate: Applied to Electric Power and Energy System Cases," in *2019 IEEE 23rd International Enterprise Distributed Object Computing Conference (EDOC)*, 2019, pp. 88–97.
- [20] L. Almgren and J. Holm Åström, "Probabilistic modelling and attack simulations on AWS Connected Vehicle Solution: An Application of the Meta Attack Language," B.Sc thesis, KTH, Stockholm, Sweden, 2019.
- [21] V. R. Kebande, S. Alawadi, F. M. Awaysheh, and J. A. Persson, "Active Machine Learning Adversarial Attack Detection in the User Feedback Process," *IEEE Access*, vol. 9, pp. 36 908–36 923, 2021.
- [22] L. Evensjö, "Probability analysis and financial model development of the MITRE ENterprise ATT&CK Matrix's attack steps and mitigations," B.Sc thesis, KTH, Stockholm, Sweden, 2020.
- [23] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," vol. 24, no. 3, 2007, pp. 44–77.
- [24] R. S. Labs. (2022, Apr) Pacu: The open source aws exploitation framework. [Online]. Available: <https://rhinosecuritylabs.com/aws/pacu-open-source-aws-exploitation-framework/>
- [25] (2022, Mar) AWS Free Tier FAQs. [Online]. Available: <https://aws.amazon.com/free/free-tier-faqs/>
- [26] (2022, Apr). [Online]. Available: <https://github.com/foreseeti/securicad-aws-collector>
- [27] C. Tafani-Dereeper. (2022, Jan) Elevate AWS threat detection with Stratus Red Team. [Online]. Available: <https://www.datadoghq.com/blog/cyber-attack-simulation-with-stratus-red-team/>

CONTEXT N – PART I

INFORMATION ENGINEERING: BIG DATA & AI

POPULAR DESCRIPTION

An AI a day keeps the doctor away

Imagine a future where doctors are no longer needed to make a simple diagnosis and your personal trainer is your smartphone. This could be a reality in as soon as 10-15 years if we utilize state-of-the-art Artificial Intelligence (AI) algorithms together with big amounts of data. AI could revolutionize our current healthcare, but may at the same time pose a risk to individual privacy.

The shortage of medical personnel is becoming an ever increasing problem. As the recent pandemic has highlighted, the amount of doctors, nurses, and surgeons are not enough. A simple solution to this dilemma is to hand out less complicated tasks to someone who doesn't need a long education, who doesn't need to be paid or be free on weekends. Who? AI! With the use of AI, diagnosing or deciding the priority of patients no longer needs to be done by educated hospital personnel but could be done by a cheap computer. This would relieve well educated doctors allowing them to focus on difficult tasks, without decreasing the quality of the healthcare given.

But AI is not only applicable within hospitals. It could also be used to improve public health by allowing individuals to monitor their own physical activities and fitness. Using only a smartphone, it will be possible to see how large portions of the day you spend sitting down, or how many steps you took last week, which could help guide you to a more active lifestyle. This more informed lifestyle comes at a cost. AI systems that make accurate predictions about you will also need large amounts of your private data. This means we will have to trust the medical community and AI engineers to respect personal integrity and keep the data private.

The future of both healthcare and personal fitness lies within the field of AI, which could help democratize expensive procedures and fitness advice as long as personal privacy is prioritized.

SUMMARY OF PROJECT RESULTS

Context N part 1 consists of a variety of different projects (projekt N1 – project N4), ranging from theoretical studies to practical implementations. The commonality of these projects is that they all revolve around big data, and data processing together with Artificial Intelligence (AI). The theoretical projects explore the foundations of machine learning, while the more practical projects use pre-existing algorithms for different applications.

The N1 project groups, N1a and N1b have investigated signal processing, step detection and Human Activity Recognition techniques using smartphone sensor data. Step detection was achieved by processing the smartphone's accelerometer sensor data through a filter and developing an algorithm to identify the steps. The groups then investigated machine learning techniques to identify activities such as walking, running, biking and climbing stairs.

To further improve the step counter, a method for ignoring sensor data that is clearly not a step, such as the act of placing the smartphone in the pocket, could be developed. It could also be interesting to develop a machine learning based step counter and see how it compares.

The N2 project groups have made a practical implementation using machine learning to predict mortality using medical data. Both project groups used the *Medical Information Mart for Intensive Care* (MIMIC) clinical database. Due to the ethical aspects associated with health care, i.e. which patient to give care to first, an important aspect of medical machine learning applications in these projects was explainability.

The N2a project group decided upon using Logistic Regression, which then was trained and tested with the dataset acquired from the MIMIC-III version of the database. This study showed how hyperparameters and feature selection affected what decisions the model made and how good the classifier became.

Project group N2b utilized MIMIC-IV instead and the data was trained by a Natural Language Processing algorithm (NLP). The most commonly occurring factors for deceased and surviving patients were also extracted. The study showed that the algorithm worked better at predicting risk factors for non-surviving patients and that the reliability of the algorithms and the database need also be considered.

A possible new project as a continuation would be to research how good classifiers you could make without considering explainability. Further research could focus on testing other statistical and machine learning methods in order to compare and possibly improve the results.

The N3 project group has studied different methods of estimating generalization error bounds for simple neural networks using mutual information. This project is theoretical in its nature and most results rely on information theoretic mathematics. Generalization bound is a measure of how well a given neural network works when it is presented with unseen data. The project group has compared three different methods to analyze the generalization error on a simple algorithm for binary classification on a dataset consisting of pictures of handwritten digits 1 and 0. The performance of the three algorithms were then compared. The results indicate that the generalization error shows how well an algorithm will perform before being deployed and thus, it could be very useful for critical applications where the margin of error is low.

The generalization bounds calculation varies significantly from method to method so there could be room for further improvement in calculating bounds. The bound is resource intensive and mathematically complex to calculate. Further research should address this problem and streamline the calculation of generalization bound as measure for algorithm performance.

Project group N4 tackled the lack of user data privacy caused by traditional machine learning methods. Traditional methods need all the raw user data to be collected on a centralized server, increasing the risk of exposing sensitive data. Using a new decentralized method called Federated Learning (FL), trained models created by the individual user devices are instead sent to a server to be combined. To simulate a real-world scenario, a testbed of Raspberry Pi:s (single board computers) was created, on which FL was implemented. The performance and accuracy of the testbed were then compared to traditional methods.

Further research could be conducted to evaluate the degree of privacy preservation of FL. Although only the models are sent to the centralized server, there is still a possibility that some training information can be derived from the models. Therefore, further improvements on the method can be made, for instance incorporating homomorphic encryption or differential privacy.

IMPACT ON SOCIETY AND ENVIRONMENT

Big Data and AI offer solutions to some previously hard-to-solve problems, as long as there is a large enough dataset and a smart enough algorithm. But AI also comes with ethical concerns regarding the environment, privacy, trust, and accountability.

The main environmental concern in this context lies in the high-power consumption that is required to train certain large machine learning models whose usefulness is uncertain. Luckily, if done properly, the model only needs to be trained once and is, therefore, a one-time energy cost. On the other hand, certain types of AI can be used to lower power consumption in some fields, and in such a case, the net gain of the environmental impact would be only positive.

One ethical aspect that has to be considered when working with large datasets and AI is that the data may reveal personal information that the user would like to remain private. This information could be used by an adversary to track an individual's movement, activities, and even infer their opinion on different matters. In the worst-case such a tool could be used by authoritarian regimes to suppress the population. However, it could also be used in more subtle ways that the user may not be aware of, like targeted advertising or workplace surveillance. On the other hand, if handled correctly, activity recognition technology could be used for positive means such as improving both public and personal health, stopping dangerous individuals before they act, and monitoring children, the elderly, and fragile people.

Another issue that AI faces is the problem of explainability. The current models are designed in a way that is often hard to interpret. Knowing the reason why the model came to a certain conclusion, can have an impact on deciding if one should trust the responding actions. This is especially relevant when working with models deployed in a medical setting where a wrong decision can be the difference between life and death, increasing the importance of accountability and trustworthiness of the models.

A machine learning system will only ever be as good as the data it was trained on, which means that if the data does not capture the full picture it may result in AI that fails to accurately represent the real world. This could have disastrous consequences if the AI is in charge of important decisions, or lead to discrimination against groups that were not represented in the training data. However, there is ongoing research on reducing bias in future machine learning systems.

With the ongoing escalations and tensions in Europe, the risk for a purely digital war is becoming more likely and AI is a powerful tool to both execute and defend against such a scenario. The risk of big data leaks that reveal economic, military, and other political strategies is ever rising and the outcome of a large attack has the potential to affect nearly all aspects of everyday modern life as well. Power plants could be sabotaged and communication networks could be cut off. The use of AI for disinformation could play a part, such as a fake recording or video of a country's leader urging its armed forces to surrender.

One important impact for individuals will be unemployment as a result of the AI workforce. Humans must adapt to doing more abstract and creative tasks that cannot be performed by AI since AI can be both more time and cost-effective. This will mean that more people need to be more educated if they want to get a job. On the other hand, AI can be used to relieve workers in certain fields where there exists a shortage of educated staff, with health care being one example.

We are also likely to interact with AI much more in the future. One example currently is robot calls, where an automated response is recorded and is played to callers and it can be used to collect some specific information needed.

With the rise in the use of smart sensors everywhere and the value that the collected data bring to businesses and governments around the world, a lot more data will likely be collected from individuals. There will likely be a need for new policies to be implemented to protect individuals from giving away any important data without consent.

To summarize, big data and AI are key enabling technologies that will impact different parts of the environment, society, and even individual lives. The scope of the projects in this context is quite diverse and the range includes applications in the health sector, sensors, and even mathematical models for reducing the bias of machine learning algorithms. The far-reaching presence of AI in future societies requires considerate planning and legislation or it could lead to massive uncontrolled societal changes, such as increased unemployment, that could create further problems.

Human Activity Recognition and Step Counter Using Smartphone Sensor Data

Gustaf Sidén and Fredrik Jansson

Abstract—Human Activity Recognition (HAR) is a growing field of research concerned with classifying human activities from sensor data. Modern smartphones contain numerous sensors that could be used to identify the physical activities of the smartphone wearer, which could have applications in sectors such as healthcare, eldercare, and fitness. This project aims to use smartphone sensor data together with machine learning to perform HAR on the following human locomotion activities: standing, walking, running, ascending stairs, descending stairs, and biking. The classification was done using a random forest classifier. Furthermore, in the special case of walking, an algorithm that can count the number of steps in a given data sequence was developed. The step counting algorithm was not based on a previous implementation and could therefore be considered novel. The step counter achieved a testing accuracy of 99.1% and the HAR classifier a testing accuracy of 100%. It is speculated that the abnormally high accuracies can be attributed primarily to the lack of data diversity, as in both cases only two persons collected the data.

Sammanfattning—Mänsklig aktivitetsigenkänning är ett växande forskningsområde som handlar om att klassificera mänskliga aktiviteter från sensordata. Moderna mobiltelefoner innehåller många sensorer som kan användas för att identifiera de fysiska aktiviteterna som bäraren utför, vilket har tillämpningar inom sektorer som sjukvård, äldreomsorg och personlig hälsa. Detta projekt använder sensordata från mobiltelefoner tillsammans med maskininläring för att utföra aktivitetsigenkänning på följande aktiviteter: stå, gå, springa, gå upp för trappor, gå ned för trappor och cykla. Klassificeringen gjordes med hjälp av en "random forest"-klassificerare. Vidare utvecklades en algoritm som kan räkna antalet steg i en given datasekvens som samlats in när användaren går. Stegräkningsalgoritmen baserades inte på en tidigare implementering och kan därför betraktas som ny. Stegräknaren uppnådde en testnoggrannhet på 99,1% och aktivitetsigenkänningen en testnoggrannhet på 100%. De oväntat höga noggrannheterna antas främst bero på bristen av diversitet i datan, eftersom den endast samlades in av två personer i båda fallen.

Index Terms—Human Activity Recognition, Step Counter, Smartphone Sensor Data, Accelerometer, Gyroscope, Random Forest.

Supervisors: Prakash Borpatra Gohain, Magnus Jansson

TRITA number: TRITA-EECS-EX-2022:167

I. INTRODUCTION

The ubiquitous adoption of sensor-rich smartphone devices in modern society has resulted in an abundance of data containing potentially valuable information about the smartphone user or the surrounding environment. Human Activity Recognition (HAR) is a field of research concerned with classifying specific activities performed by humans via various

kinds of sensors, such as accelerometers, gyroscopes, GPS, or optical devices like RGB and depth cameras. By combining HAR with smartphone sensor data, it is possible to create a cheap and efficient method for analyzing physical activities in everyday life [1].

One area that could benefit from easily accessible, non-vision-based activity recognition is the healthcare and fitness sector, where it can be used to track the daily physical activities of both individuals and larger cohorts. It allows for a data-driven and quantitative approach to analyzing activity patterns and can reveal information such as how much time an individual spends sitting down daily or what portion of the population runs at least once a week. These insights can then be used to provide actionable advice on improving the physical well-being of individuals.

The purpose of this project is to perform Human Activity Recognition from smartphone sensor data using machine learning, as well as develop an algorithm for counting the number of steps taken in a given walking data sequence. Although step counting can be viewed as a more quantitative subset of HAR, these are in practice two completely different tasks performed using two different methods. This project could therefore be considered two separate, smaller projects within the same field of research.

More specifically, the HAR portion of the project is concerned with classifying the following activities from the smartphone's accelerometer and gyroscope data: standing, walking, running, ascending stairs, descending stairs, and biking. The classification is performed using a machine learning method called random forest. The step counter uses only the accelerometer data and is implemented using a technique that will be described in this paper.

II. THEORY

A. Step Counting

Step counting algorithms that leverage smartphone sensor data tend to primarily analyze the accelerometer data, although this is occasionally complemented by other sensors such as the gyroscope or magnetometer [2] [3] [4]. Since the step counter described in this paper uses the accelerometer, the focus of this section will be accelerometer-based techniques.

Since walking is a periodic, oscillating motion, the output of the accelerometer could be roughly modeled as a sinusoidal obscured by high-frequency stochastic noise.

$$Y(n) = X(n) + V(n) \quad (1)$$

where $Y(n)$ is the observed signal, $V(n)$ is an unknown noise, and $X(n)$ is the dominant motion of the walking pattern,

modeled as

$$X(n) = A \sin(2\pi fn + \Phi), \quad (2)$$

where A is the amplitude of the acceleration, f is the walking frequency, and Φ is the walking phase.

If $X(n)$ can be recovered from $Y(n)$, counting the number of steps in a given time frame is in principle trivial and could be done by counting some periodic characteristics of $X(n)$, such as the peaks in the signal. However, in practice, this task is more difficult than that since the acceleration amplitude A is not constant over time and will depend on factors such as shoe stiffness, walking surface, walking style and walking speed.

The problem of step counting can thus be broken down into two distinct parts; (i) isolating the dominant motion $X(n)$ from the raw data sequence $Y(n)$, and (ii) choosing a threshold $T < A$ such that all peaks in $X(n)$ above T are steps and no peaks below T are steps, where A is the amplitude from (2). Generally, the solution to these two sub-problems is where step counting algorithms differ.

B. Random Forest

A *random forest* is a type of ensemble machine learning model based on *decision trees*. A decision tree is also a machine learning model that can be seen as a piece-wise constant approximation of the underlying function. Decision trees make inferences by evaluating a sequence of *if-then* conditionals for the features, which is equivalent to traversing a tree structure that branches at each *if-then* conditional and where the leaf nodes are the final decisions [5].

Although decision trees can be useful in many applications, especially in simple classification problems, they are prone to overfitting and do not tend to generalize well beyond the training data. However, it can be shown that by averaging the results from multiple, slightly different decision trees, the overfitting can be drastically reduced while still maintaining the predictive power [6]. The reasoning behind this is that each tree will be overfit in a slightly different way, so the average of each overfit value is closer to the real value. Variations in the decision trees are introduced by injecting randomness into the training process for each tree. The randomness affects the data points that the tree has access to, as well as the features it looks at to make decisions [5]. Fig. 1 shows a visualization of what a random forest may look like.

Although there are variations of the algorithm, the most common implementation is from the original paper from 2001 by Leo Breiman [6]. Its implementation will be briefly described in this section.

Given a training data set containing N data points, each with M features, the algorithm can be broken down as follows:

1) *Bootstrapping*: Randomly sample $n < N$ data points from the data set, with replacement (a sampled data point could be sampled again). This subsampling process is called *bootstrapping* or *bagging*.

2) *Random subspace*: Randomly select $m < M$ features, without replacement as it would be redundant to include a feature multiple times. Together with the previous step, n and m represent a portion of the training set.

3) *Training estimators*: Each decision tree is called an estimator. For each estimator, train on a separate set of n and m examples.

4) *Perform inference*: If T is the number of estimators, then evaluating the forest yields T predictions. To reach a final decision, a majority vote is cast. The class predicted by the most estimators is the prediction of the random forest.

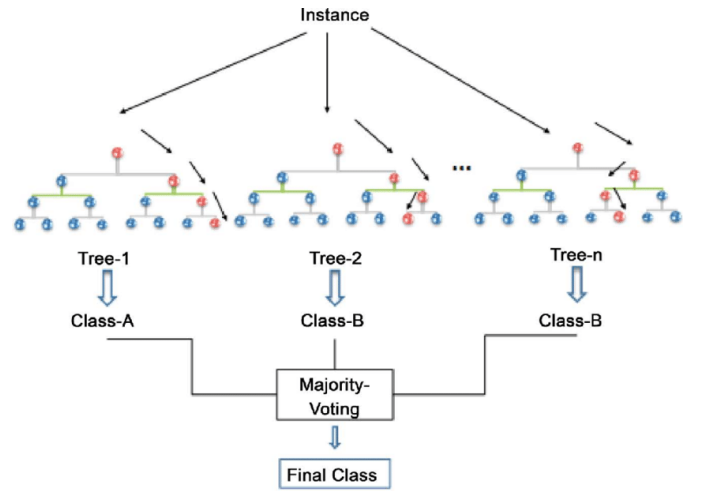


Fig. 1. Visualization of a random forest. Sourced from [7], licensed under Creative Commons 4.0 International.

III. METHOD

A. Data Collection and Preprocessing

1) *Step Counter*: For the step counting algorithm, the *MATLAB Android* application was used to log the smartphone accelerometer data in the X, Y, and Z directions. Due to a memory limitation in the application, a sample rate of 10 Hz was used. The smartphone was then placed in the breast pocket, front pocket, or rear pocket and the user walked 30, 50, or 100 steps and stopped the recording. The number of steps taken was noted upon saving for future reference. In total, 46 walking sessions were collected.

To process the log files, the desktop application *MATLAB R2022a* was used. Instead of processing the acceleration data on a per-axis basis, the magnitude of the acceleration vector was used. The first and last five seconds were cut from each log to remove the data from when the user held the phone to start or stop sensor logging, and 15 samples with an amplitude of 10 were added to the beginning for the filter to stabilize. Data from each walking session was then visualized in a graph to ensure that the data looked nominal, see figure 2.

2) *HAR*: For the HAR, more sensor data at a higher sample rate is desired. Therefore, another application called *Sensor Logger* by Kelvin Tsz Hei Choi for the *Android* operating

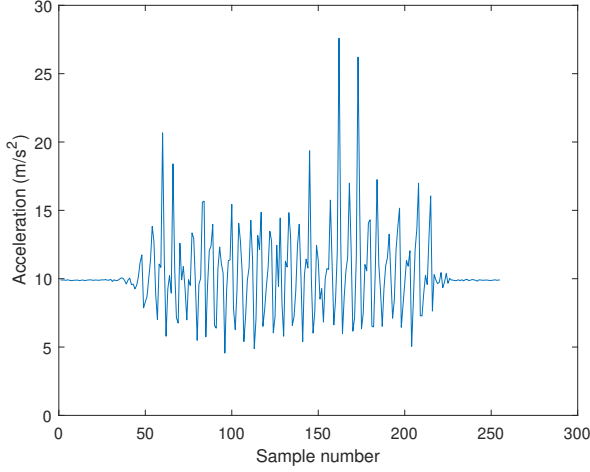


Fig. 2. Example of acceleration magnitude data sequence after trimming and adding initial samples.

system was used to log sensor data at a sample rate of 50 Hz. The accelerometer data in the X, Y, and Z directions and orientation data were stored in a Comma Separated Values (CSV) file. The orientation data were collected as roll, pitch, and yaw, which span the range from $-\pi$ to $+\pi$ in radians, giving a complete representation of the current orientation. Fig. 3 shows the standardized coordinate system for acceleration and orientation for a smartphone. With the smartphone in the front pocket, the user then performed one of the following activities: biking, ascending stairs, descending stairs, running, standing, or walking. After the activity, the user saved the log and named it according to the activity that was performed.

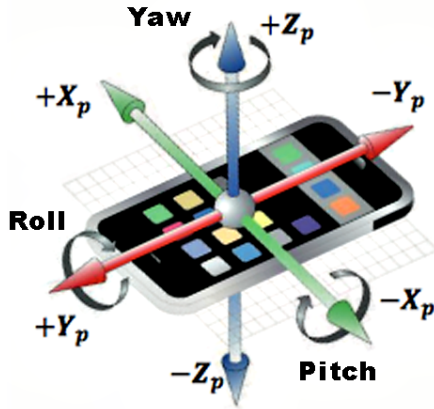


Fig. 3. Visualization of the coordinate system for acceleration and orientation in a smartphone. Sourced from [8], licensed under Creative Commons Attribution 3.0 Unported and adapted to this report with orientation labels.

The data were then processed using *MATLAB R2022a*, trimming away the first and last five seconds of each file when the user was expected to handle the phone, as well as removing portions where the activity was not performed. Table I shows the number of samples captured for each activity. Due to an unknown reason, large chunks of the data got corrupted on one of the phones used and had to be cut, resulting in less data

than anticipated. Figure 4 and 5 may be used as a reference for what nominal data looked like.

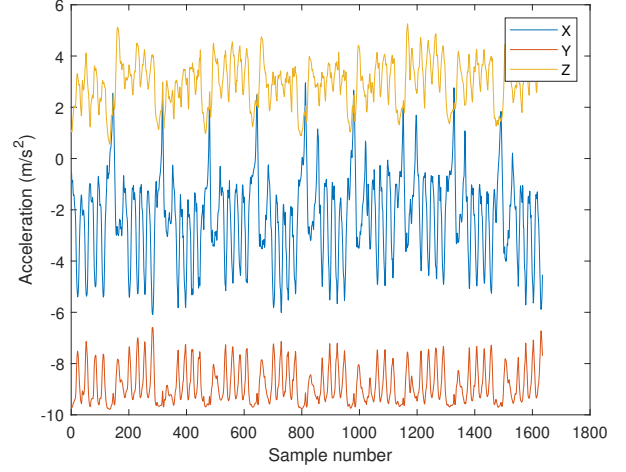


Fig. 4. An example of nominal HAR data, acceleration only.

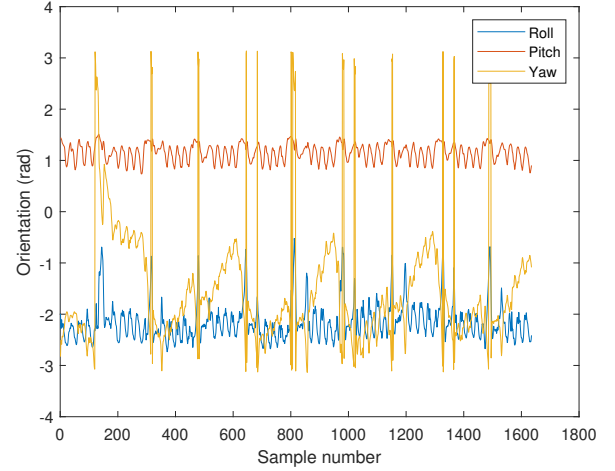


Fig. 5. An example of nominal HAR data, orientation only.

TABLE I
THE NUMBER OF ACTIVITY DATA SAMPLES THAT WERE AVAILABLE

Activity	Samples
Biking	668125
Ascending stairs	12226
Descending stairs	11391
Running	162046
Standing	36104
Walking	38196

All the cut files for each activity were then merged into one new CSV file for each activity to be used in the final preprocessing step where the combined data for each activity is read into smaller segments consisting of 250 samples, representing five seconds of time. This time frame was chosen as it would include multiple oscillations for each locomotion activity. For each segment, the following features were calculated: mean,

standard deviation, maximum, minimum, range, median, and root-mean-square (RMS) for each axis of both the acceleration and the orientation. Additionally, the activity labels were one-hot encoded in a vector. In total, each data point included 42 features excluding the one-hot vector.

To maximize the utility of the raw data, as well as to balance the number of data points for the different activities, a kind of random oversampling was used when selecting segments. If the segments were chosen such that any one sample could only belong to a single segment, then the resulting segmentation would be an evenly distributed sequence. However, since the data sequence is a repeating pattern, any one segment is likely to look similar to another segment. Therefore, it could be reasonable to allow a sample to belong to multiple segments, as it would not drastically alter the pattern that segments tend to look similar for a given activity. The result would be many more unique segments, with the drawback being that samples belonging to a segment are not necessarily exclusive to that segment.

Random oversampling was performed by randomly selecting a sample in the data sequence for the activity and extending 250 samples to create a segment. This process could be repeated as many times as desired to yield any number of segments, although at some point, the generated segments would be identical to some previous segment, increasing the risk of overfitting the model. It was decided that 1000 segments (resulting in 1000 data points) for each activity would be used as an initial number of data points. If the model were to display signs of overfitting, this number could be adjusted. Fig. 6, and Fig. 7 show a visualization of the segmentation process, comparing sequential segmentation to random sampling. Table II and Table III show the number of segments generated for each activity using the two methods.

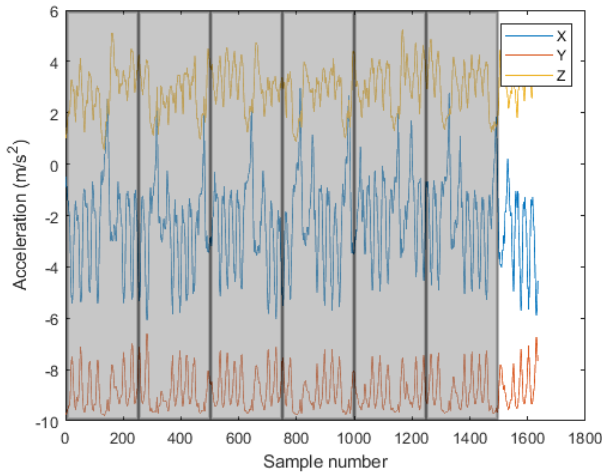


Fig. 6. Visualization of sequential segmentation. Each sample belongs to only one segment, limiting the number of possible segments.

The data that were to be used for testing were split from the rest of the data before the oversampling to ensure that the algorithm was evaluated on unseen data. Because of the small amount of data for certain activities before oversampling, the number of test segments for each activity was limited to ten.

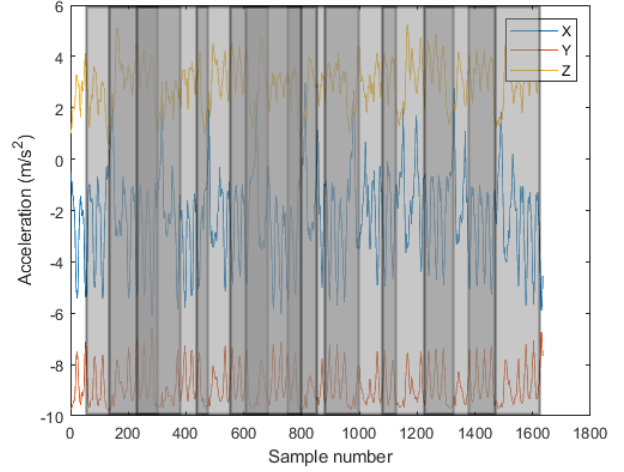


Fig. 7. Visualization of randomly sampled segments, where each sample can belong to multiple segments. This shows 11 unique segments, although not every sample is exclusive to one segment.

TABLE II
THE NUMBER OF SEQUENTIAL SEGMENTS THAT ARE POSSIBLE FOR EACH ACTIVITY

Activity	Segments
Biking	2672
Ascending stairs	48
Descending stairs	45
Running	648
Standing	144
Walking	152

B. Design of Step Counting Algorithm

As described in part A of the theory section, step counting can be divided into two parts: extracting the dominant motion from the raw data and counting the peaks in the signal that corresponds to the steps.

Since the noise in the data is generated primarily by processes that have a higher frequency than the walking frequency (such as the smartphone vibrating in the pocket), it is reasonable to assume that most of the energy in the noise is located at frequencies higher than the walking frequency. Therefore, a low-pass filter with a cutoff frequency around the walking frequency was used to filter the signal and reveal the dominant walking pattern.

The filter was of type *Butterworth* with order $n = 3$ and an initial cutoff frequency $f_c = 2.3$ Hz. The filter cutoff frequency was chosen in such a way that the filter would dampen frequencies higher than a regular walking frequency (≈ 2 Hz) while leaving the desired frequencies relatively unchanged. However, in a later optimization step, f_c was further improved, which is explained later in this section.

Peak counting was performed by comparing sample n with samples $n - 1$ and $n + 1$. If sample n is greater than both the previous and the next sample, it is a peak. However, even the filtered signal contained many smaller peaks that were not steps. To isolate the peaks that represented steps, a threshold was introduced, where any peak above the threshold

TABLE III
THE NUMBER OF SEGMENTS AFTER OVERSAMPLING

Activity	Segments
Biking	1000
Ascending stairs	1000
Descending stairs	1000
Running	1000
Standing	1000
Walking	1000

represented a step.

This threshold had to change dynamically with the data since peak values could differ significantly between different data sequences depending on numerous factors, such as the walking surface, loose or tight pockets, or worn shoes.

Through experimentation, it was found that the optimal threshold (the threshold that included all step peaks and excluded all non-step peaks) for a given data sequence was a function of the mean peak value in the sequence. A higher mean peak value resulted in a threshold that should be higher and vice versa, although the threshold still had to be calibrated additionally for each data sequence. More specifically, let T be the threshold, μ the mean peak value, and K a calibration factor. By inspecting how T should change with μ for a given data sequence, it was found that T depends on μ as

$$T(\mu) = K\mu. \quad (3)$$

Furthermore, by plotting a small, randomly selected set of manually calibrated thresholds, it was found that the calibration factor K was itself a first-degree polynomial and a linear function of μ . The manual calibration was performed by finding a value for K that satisfied (ii) in the theory section. Therefore, K was given as

$$K(\mu) = a\mu + b, \quad (4)$$

where a and b are real-valued constants. Thus, the threshold $T(\mu)$ is a second-degree polynomial of μ . The constants a and b can be calculated by performing linear regression on the data set of manually calibrated thresholds. However, since this data set had been created by manually finding the optimal thresholds, a more automated method for calculating the optimal constants a and b was desired.

Since a and b remain constant and independent of the data, the optimal value should be chosen with respect to the entire data set. In other words, a and b were chosen such that the mean square error (MSE) across the available data was minimized.

Let Z_i denote the ground truth number of steps in the data sequence i , \hat{Z}_i the predicted number of steps according to the algorithm, and n the total number of data sequences available in the training data. Then, the following sum gives the MSE across the available data:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Z_i - \hat{Z}_i)^2. \quad (5)$$

Another crucial factor that remained constant was the filter cutoff frequency, which could be optimized in the same

manner. Let $F(a, b, f_c)$ be the function that returns (5) when evaluated with the given parameters for a , b and f_c . The problem can then be described as finding the parameters a , b and f_c that minimize F .

The optimization was performed using the Nelder-Mead simplex algorithm, a numeric method for finding the minimum value of a given multidimensional objective function. The implementation was part of the MATLAB library, which uses the implementation described in [9]. To simulate a more realistic use case with unseen data, an 80/20 training/testing split was performed, so that the parameters were not optimized for the unseen data.

Additionally, a graphical user interface (GUI) was developed to improve ease of use of the system. The user could upload their data sequence by pointing to its storage location, and the program would then calculate the number of steps using the algorithm that was described in this section, as well as give a visual indication of where the steps were detected in the signal.

C. HAR Using Random Forest

The Human Activity Recognition of the locomotion activities was performed using an implementation of a random forest classifier in the Python machine learning library *sci-kit learn*. This implementation is an improved version of the algorithm described in the theory section, where the improvement stems from averaging the probabilistic predictions of each tree, instead of doing a majority vote.

The random forest was built without asserting any constraints on the maximum depth or maximum number of leaf nodes. The number of trees in the forest was set to $n_{\text{estimators}} = 100$. This value was found to be sufficient, as the accuracy plateaued and increasing it further provided no additional gains. The decision trees were built using bootstrapped samples from the data, as described in the theory section. Both bootstrapping, and feature selection used the same randomization seed for reproducible results, arbitrarily chosen as number 42.

As described in the preprocessing step, the data set was enhanced using a type of random oversampling to achieve an even distribution of data points for the different activities. The oversampled data set was used for training, and the testing was done with a separate data set that was not part of the original data set that was oversampled. In this way, it was ensured that testing data had not been seen by the algorithm during training.

D. Performance Evaluation

1) *Step Counter*: The error for the step counter was measured by mapping the error in the number of steps to the range $[0, 1]$ and computing the average for all data sequences. Let Z_i denote the ground truth number of steps for a data sequence i , \hat{Z}_i the predicted number of steps, and n the number of data sequences in the testing data. Then, the error E_{step} was computed as

$$E_{\text{step}} = \frac{1}{n} \sum_{i=1}^n \min \left(\frac{|Z_i - \hat{Z}_i|}{Z_i}, 1 \right), \quad (6)$$

and the accuracy as $1 - E_{step}$. Since \hat{Z}_i can be arbitrarily large, it was necessary to assert a limit where the error was considered maximal. The natural choice was to define the maximum error where the algorithm missed all steps, $\hat{Z}_i = 0$. In order to retain reflection symmetry around the point where the algorithm detected all steps, $\hat{Z}_i = Z_i$, the upper limit would thus need to be $\hat{Z}_i = 2Z_i$. This definition ensures that the error for a given data sequence is mapped to the range $[0, 1]$, where an error of zero means that all steps were detected, and an error of one means that either no steps were detected, or that the number of detected steps was greater than twice that of the ground truth number of steps. Since E_{step} is the average for all data sequences, it too will be limited to the range $[0, 1]$.

2) *HAR*: The error for the HAR classification was measured as the relative error between the number of correctly predicted labels and the number of tested labels. Let \mathbf{V}_i be the one-hot encoded vector representing the label for testing data point i , $\hat{\mathbf{V}}_i$ the predicted label, and n the number of testing data points. Then, the error E_{HAR} was computed as

$$E_{HAR} = \frac{1}{n} \sum_{i=1}^n \frac{\|\mathbf{V}_i - \hat{\mathbf{V}}_i\|}{\sqrt{2}}, \quad (7)$$

and the accuracy as $1 - E_{HAR}$, where $\|\cdot\|$ represent the L2-norm of the resulting vector, calculated as $\|\mathbf{V}\| = \sqrt{\sum_{k=1}^m (V_k)^2}$ where V_k is vector component k of \mathbf{V} that has m components. Since \mathbf{V} is a one-hot encoded vector, only one of its axes will be nonzero, and that axis will be one. Thus, the summand in equation (7) can be seen as a piece-wise function that returns zero if $\mathbf{V}_i = \hat{\mathbf{V}}_i$ and one if $\mathbf{V}_i \neq \hat{\mathbf{V}}_i$ since the norm of the resulting vector in the latter case will always be $\sqrt{2}$. Therefore, E_{HAR} is also mapped to the range $[0, 1]$, where zero represents that all of the predicted labels were correct, and one that none of the predicted labels were correct.

IV. RESULTS

A. Step Counter

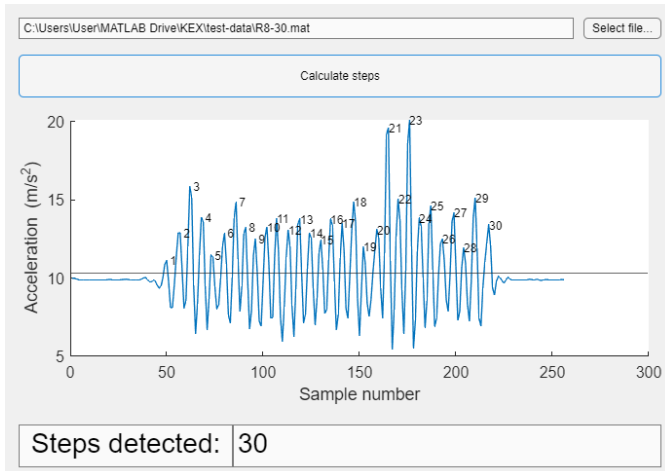


Fig. 8. Step counter GUI. The detected steps are visually located at the peaks, and the total number of steps in the sequence is presented at the bottom.

Using the initial values for the parameters a , b and f_c , the step counter achieved an accuracy of 97.1%. However, after performing the optimization step described in the method, the accuracy improved to 99.1%. The initial parameter and the optimized values for the parameters can be found in table IV. Figure 8 shows the GUI that was developed alongside the algorithm. The GUI shows a data sequence of 30 steps that were fed into the algorithm, with the correctly identified steps located at the peaks.

TABLE IV
THE STEP COUNTER PARAMETERS BEFORE AND AFTER OPTIMIZING.

Parameter name	Initial parameters	Optimized parameters
a	-0.050	-0.053
b	1.50	1.48
f_c	2.30	2.26

B. HAR

The Human Activity Recognition achieved an accuracy of 100%, and therefore managed to correctly label every activity in the testing data. Figure 10 shows the confusion matrix, where the special case of 100% accuracy reduces to an identity matrix. A feature importance analysis showed that acceleration was significantly more important than orientation in determining activity, and that the X-axis had the highest importance. The importance of each feature is shown in figure 9. Rounded to the closest integer number, 70% of the importance is distributed on the acceleration, and the remaining 30% on the orientation data. Figure 11 shows an example of a single decision tree from the forest.

V. DISCUSSION

As seen in the results, both the step counter and the random forest achieved remarkably high accuracy on their respective tasks. This can be partially explained by the lack of variety in the data. Since all data were collected by no more than two persons, the data set fails to account for individual variations such as locomotion style or shoe variety. The data that is used for testing will therefore be similar to the data used during training.

However, it is also likely the case that identifying locomotion activities is simply a task that is well suited for a random forest classifier. The features used to characterize the signal are different enough between each activity that the decision trees in the forest can find meaningful splits when constructed, resulting in the entire forest being able to accurately predict the activity.

Furthermore, because of the oscillating nature of locomotion, a randomly selected segment in the data is likely to look like any other segment in the data since it is a repeating pattern. This means that during testing, the provided unseen data are likely to be similar to the data used during training, thus increasing the predictive power even further for this specific use case. The accuracy would certainly be lower if there was more variation in the data set, although it is believed that

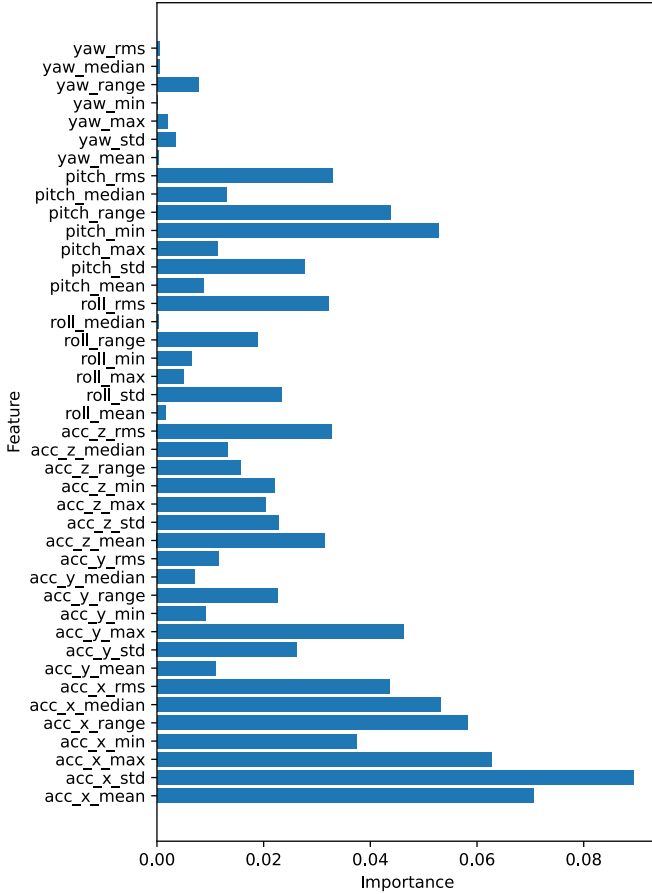


Fig. 9. The importance of the evaluated features. The sum of all importance is one, and each importance value represents how much that feature contributed to the final decision compared to all the other features.

the algorithm would still achieve very good testing accuracy because of the aforementioned reasons.

One issue when doing random oversampling to enhance the data set is that the risk of overfitting increases. Due to the homogeneity of the data used, it may be the case that the random forest has been slightly overfit to the data set used and therefore generalizes poorly. This would require additional, different data to confirm. However, as explained in the theory, one of the main advantages of random forests is that they are quite resistant to overfitting, and this issue may, in that case, be solved by providing more varied data for training.

Random forests are also generally considered robust and accurate classifiers, albeit quite expensive from a computational perspective. Because of this, they are not usually deployed in real-time use cases. With that in mind, one could argue that this algorithm is unnecessarily powerful for the task of classifying locomotion, since it is evident that this is an easy task compared to many other machine learning applications. A suggestion for future work is therefore to develop a handcrafted technique for this task and compare it to the random forest classifier.

As shown in the feature importance analysis of the random forest classifier, many features are redundant or have a negligible impact on the outcome. Another avenue for further

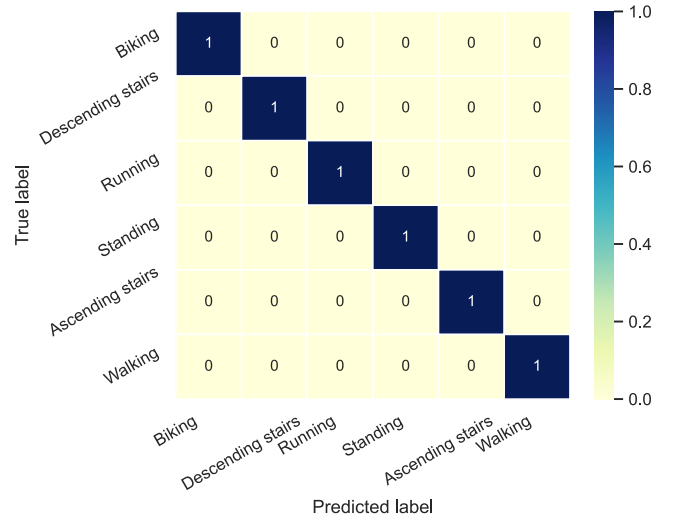


Fig. 10. The confusion matrix for the HAR classification. Since each activity was correctly labeled, the matrix is an identity matrix.

research is to see how many features can be removed without adversely affecting the accuracy or to find some different feature that manages to achieve an even higher importance. This would mean the forest could be made smaller and cheaper to evaluate.

One reason that the orientation data was less useful than the acceleration data could be that it includes discontinuous data where the orientation flips from a full rotation back to zero, making it hard to interpret. A possible solution to this is to encode the orientation data in a format that handles rotations continuously, such as sine and cosine. This would be a more accurate representation of the meaning of the data in reality and would likely result in it being more useful for determining the activity.

For the step counting algorithm, the same reasoning about the quality of the provided data applies. However, it seems reasonable that the algorithm would still perform well even on more varied walking data if the optimization step was rerun with the new data.

As described in the method, it was found that the optimal threshold was related to the mean peak value via a second-degree polynomial. The reason for this relationship was not explored further in this project, however it could be an interesting aspect to investigate in a future project. Another aspect to explore could be if there exists some better variable relating the optimal threshold to the observed signal, such as the variance.

Regarding finding the optimal coefficients for the polynomial, the process was automated only after finding good initial values, which required performing a linear regression on a set of manually calibrated threshold values. However, the same type of optimization could be used to generate those data points automatically for all data sequences, which would remove every manual step of the process and allow the algorithm to adapt dynamically to any data set. The Nelder-Mead simplex algorithm would be unnecessarily expensive

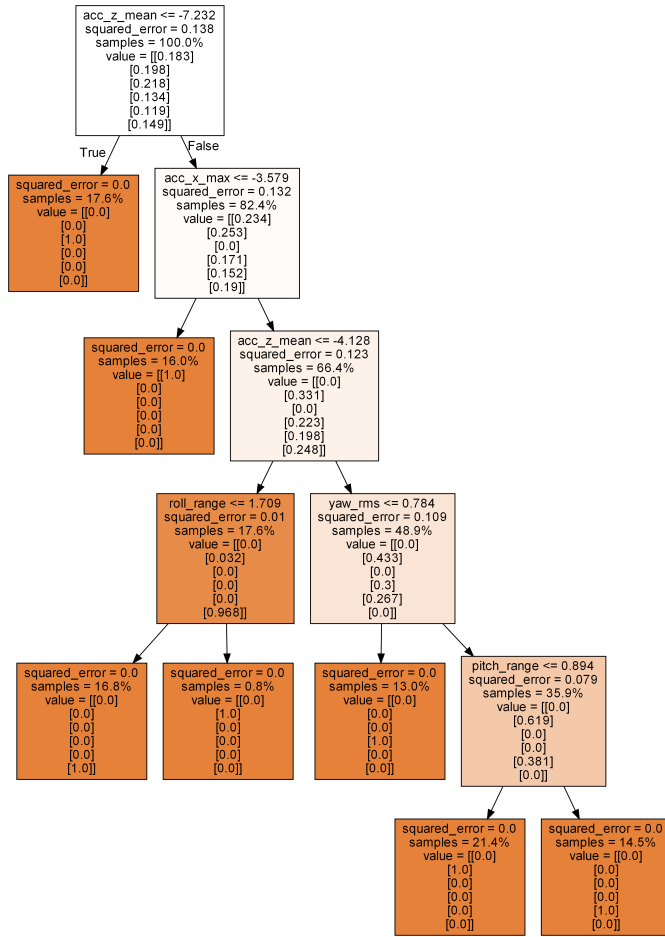


Fig. 11. Visualization of one decision tree in the random forest. Each block represents a branching node or a leaf node. The features being evaluated, as well as the outcome of the evaluation, are shown in the block. The color represents how certain the tree is of a specific outcome.

for this - any decent numerical minimization method should suffice. The process would involve minimizing the error for a given data sequence rather than the entire data set to find the threshold where all steps are included. These threshold values could then be used in the same way as described previously to perform a linear regression and find the coefficients.

The results also showed that the optimized parameters were very close to the initial values. This is most likely the result of the optimization process finding a local minimum close to the initial value that is “good enough”, but not guaranteed to be the global minimum. This line of reasoning is supported by experiments that were performed during the method, which showed that if the initial values changed sufficiently, the optimized values that were found also changed. Therefore, there exist multiple local minima and the optimization function will converge to one that is not guaranteed to be the global minimum.

Although the difference between the optimized and initial parameter values may seem insignificant, the result on the achieved accuracy was not: 2% increase from 97.1% to 99.1% is enough to justify the additional steps.

VI. CONCLUSION

The aim of this project was to develop a Human Activity Recognition (HAR) system to identify the following human locomotion activities: standing, walking, running, ascending stairs, descending stairs, and biking, as well as a step counter for the walking activity. Both tasks were to be solved using only smartphone sensor data. The HAR was performed using a random forest classifier, and the step counter used an algorithm that - as far as the authors are aware - contains some novel elements.

The HAR achieved an astonishing accuracy of 100%. This is likely partially due to the lack of variety in the data set, as the data were collected by only two persons. However, it is also an indication that random forest classifiers are suitable for the task at hand, and that the task itself is not difficult. While a more varied data set would undoubtedly result in lower accuracy, the random forest classifier would likely still perform well.

The step counter achieved a very high accuracy of 99.1%. While it too suffers from the same lack of data variety, it is believed that it would still perform well with new data since the algorithm includes a component that adapts to the data set. Some suggestions for future work and improvements to the step counting algorithm and the HAR system are provided in the discussion.

In summary, it has been shown that smartphone sensor data can be leveraged to achieve accurate Human Activity Recognition for locomotion tasks using a random forest classifier, as well as for counting the number of steps taken using the method described in this paper.

ACKNOWLEDGMENT

The authors want to thank their supervisor Prakash Borpatra Gohain for supporting the group throughout the project and providing feedback, as well as their supervisor Magnus Jansson for providing the opportunity to work on this project.

REFERENCES

- [1] O. C. Ann and L. B. Theng, “Human activity recognition: A review,” in *2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014)*, 2014, pp. 389–393.
- [2] Y. Huang, H. Zheng, C. Nugent, P. McCullagh, S. M. McDonough, M. A. Tully, and S. O. Connor, “Activity monitoring using an intelligent mobile phone: A validation study,” in *Proceedings of the 3rd International Conference on Pervasive Technologies Related to Assistive Environments*, ser. PETRA '10. New York, NY, USA: Association for Computing Machinery, 2010. [Online]. Available: <https://doi.org/10.1145/1839294.1839306>
- [3] W. W. Myo, W. Wettayaprasit, and P. Aiyarak, “A more reliable step counter using built-in accelerometer in smartphone,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 12, pp. 775–782, 11 2018.
- [4] M. Khedr and N. El-Sheimy, “A smartphone step counter using imu and magnetometer for navigation and health monitoring applications,” *Sensors*, vol. 17, p. 2573, 11 2017.
- [5] A. Müller and S. Guido, *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media, Incorporated, 2016. [Online]. Available: <https://books.google.se/books?id=qjUVogEACAAJ>
- [6] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, pp. 5–32, 2001.
- [7] Y. Fu, “Combination of random forests and neural networks in social lending,” *Journal of Financial Risk Management*, vol. 06, pp. 418–426, 01 2017.

- [8] Z. Zhou, "Headsup : Keeping pedestrian phone addicts from dangers using mobile phone sensors," *International Journal of Distributed Sensor Networks*, vol. 2015, pp. 1–9, 05 2015.
- [9] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder–Mead simplex method in low dimensions," *SIAM Journal on Optimization*, vol. 9, no. 1, pp. 112–147, 1998. [Online]. Available: <https://doi.org/10.1137/S1052623496303470>

Step Counter and Human Activity Recognition Using Smartphone IMUs

Max Strandell and Anton Israelsson

Abstract—Fitness tracking is a rapidly growing market as more people desire to take better control over their lives. And the growing availability of smartphones with sensitive sensors makes it possible for anyone to take part. This project aims to implement a Step Counter and create a model for Human Activity Recognition (HAR) to classify activities such as walking, running, cycling, ascending and descending stairs, and standing still, using sensor data from handheld devices. The Step Counter is implemented by processing acceleration data and finding and validating steps. HAR is implemented using three machine learning algorithms on processed sensor data: Random Forest (RF), Support Vector Machine (SVM), and Artificial Neural Network (ANN). The step counter achieved 99.48% accuracy. The HAR models achieved 99.7%, 99.6%, and 99.5% accuracy on RF, ANN, and SVM, respectively.

Sammanfattning—

Aktivitetsspårning är en snabbt växande marknad när fler människor önskar att ta bättre kontroll över deras liv. Den växande tillgängligheten på smartphones med känsliga sensorer gör det möjligt för vem som helst att delta. Detta projekt siktar på att implementera en stegräknare samt skapa en modell för mänsklig aktivitetsigenkänning (HAR) för att klassificera aktiviteter såsom att promenera, springa, cykla, gå upp eller ner för trappor och stå stilla, med användning av sensordata från handhållna enheter. Stegräknaren implementeras genom att bearbeta accelerationsdata och hitta samt validera steg. HAR implementeras med hjälp av tre maskininlärningsalgoritmer på bearbetad sensordata: Random Forest (RF), Support Vector Machine (SVM) och Artificial Neural Network (ANN). Stegräknaren uppnådde en noggrannhet på 99.48%. HAR-modellerna uppnådde en noggrannhet på 99.7%, 99.6% samt 99.5% med RF, ANN och SVM.

Index Terms—Step Counting, Human Activity Recognition, IMU, Smartphone

Supervisors: *Prakash Borpatra Gohain and Magnus Jansson*

TRITA number: TRITA-EECS-EX-2022:168

I. INTRODUCTION

Mobile devices are now used everywhere, each with a multitude of sensors tracking the most routine parts of your daily life. In its basest form, these sensors can not tell anything about what you are doing, but through some processing, this data can be turned into a functional step counter and Human Activity Recognition (HAR) model.

The strong correlation between a sedentary lifestyle and obesity, cancer, poor cardiovascular health, and many other health risks highlights the need for physical activity. Being able to track the levels of activity with a smartphone, a device many people carry with them wherever they go, allows

individuals and medical professionals to monitor, encourage, and provide advice on daily physical activity [1].

In this project, we develop a step counter using acceleration data from smartphone Inertial Measurement Units (IMUs), filtering and further processing the data. Furthermore, we examine three different machine learning algorithms and compare their accuracy for use in HAR.

Some methods of step counting include using adaptive filtering based on magnetometer and gyroscope data [2] and counting peaks of acceleration or applying auto-correlation to acceleration data [3]. This project aims to develop a step counter using a fixed filter and a method of peak counting and validating.

Previous work such as [4] and [5] have implemented HAR models using smartphone data and applying it to the Support Vector Machine (SVM) and Artificial Neural Network (ANN) machine learning algorithms. This project aims to extend the activity recognition to three algorithms: SVM, ANN, and Random Forest (RF).

The RF, SVM, and ANN machine learning algorithms are selected and trained on acceleration, angular velocity, and linear acceleration data to be able to distinguish between six common activities: walking, running, cycling, standing still, ascending and descending stairs. In developing the HAR model, we collect data using the Android app AndroSensor [6]. The step counter partially uses our data along with the existing Oxford Step Counting dataset [7], allowing for more data to test our algorithm to help improve it.

In Section II we will describe how we collected and processed our data, counted steps, and applied the different machine learning algorithms. In Section IV we present our results. In Sections V and VI we present a discussion of our results along with a conclusion.

II. BACKGROUND

A. Peak Counting

During normal walking motion, the smartphone will experience acceleration that can be represented as a sine wave, with a magnitude and period proportional to the walking speed. As such it will have repeated maximum and minimum values, one maximum and minimum together will compromise a step. By counting these, it is possible to estimate the number of steps taken.

B. Butterworth Lowpass Filter

A Butterworth lowpass filter (BLF) is used to clean up noise from the sensor and smaller movements not originating from larger movements such as walking. This filter has been shown

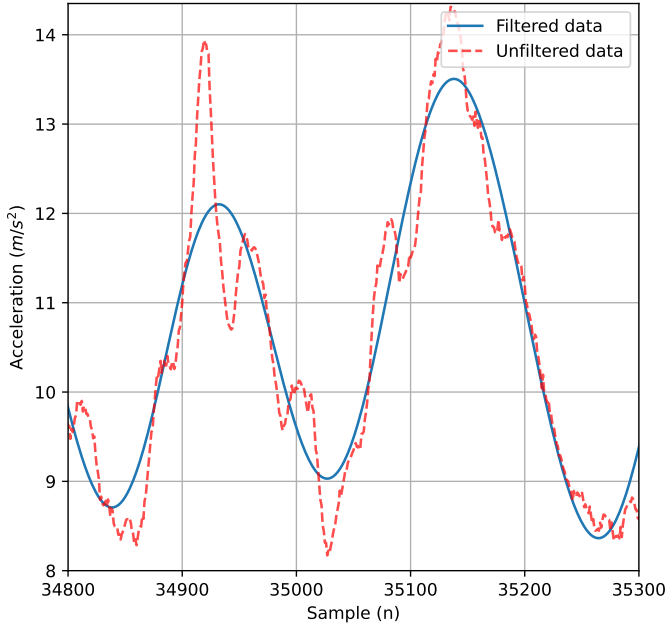


Fig. 1. Visualizing the effects of the BLF during motion, all quick variations in the data caused by signal noise are removed with the filter.

to be effective in filtering biomechanical signals by Crenna et. al. [8]. The cutoff frequency for this filter has been chosen to be 3 Hz as the usual walking frequency is around two steps per second, as determined by [9]. The effect of the filter is shown in Fig. 1.

C. Madgwick Filter

The Madgwick Filter is an absolute orientation filter that can find the direction of the device in relation to the surface of the Earth such that the z -axis is always parallel to gravity. The orientation is calculated using the accelerometer and gyroscope as proposed by Sebastian Madgwick [10], and implemented in Python using the Madgwick filter from the AHRS module [11].

D. Machine Learning Models

Random Forest (RF): The RF algorithm is a machine learning algorithm based on decision trees [12]. A decision tree can in its simplest form be described as taking input and some conditions, and based on if the conditions are met, giving an output or prediction. For example, to identify an animal the decision tree is given the condition “The animal has wings”. If it does, the decision tree predicts that the animal is a bird, and if not it is a fish. Having more and more trees (hence the name “forest”) allows for more specific conditions, which in this example could be used to identify the exact type of bird or fish. Shuffling these conditions randomly (hence the “random” part) reduces the sensitivity of the model and prevents overfitting.

Artificial Neural Network (ANN): As described by [13], the ANN is modelled to imitate the information processing of the human brain, and is comprised of layers of linear predictors referred to as nodes. In its simplest form, a neural

network consists of two layers, an input, and an output layer consisting of n and m nodes. Every node of the input layer is connected to each node of the output layer, the strength of each connection is called the weight. A network consisting of more than two layers has one or several layers, referred to as hidden layers, in between the input and output. In this case, all the input nodes are connected to all the nodes of the hidden layer. For example, if there are two hidden layers A and B, the input nodes are connected to the nodes of A, the nodes of B are connected to the output, and so on with more layers. Each node in each layer has its own weight, and during training, the values of these weights are tuned in order for the output to give the right results.

Support Vector Machine (SVM): The SVM algorithm is used to classify data by separating it by drawing a hyperplane between the data [14]. For example in \mathbb{R}^2 it separates the data by drawing a line, as seen in Fig. 2. In \mathbb{R}^3 it would separate data using a plane, and in higher dimensions a hyperplane. The algorithm uses the data points as vectors that support it in placing and rotating the plane to get the maximum amount of separation between classes, a high separation leads to a low risk of misclassifying data.

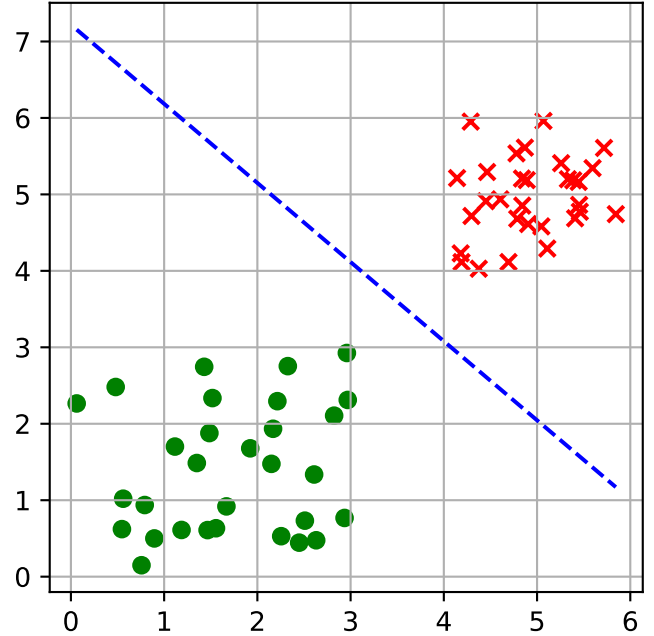


Fig. 2. An example of the SVM algorithm classifying data in \mathbb{R}^2 . It fully separates two types of data, circles, and crosses, and is able to correctly determine the class of each, not mistaking a circle for a cross or vice versa.

III. METHODOLOGY

A. Data Collection

1) Our Dataset: Data was collected using a Moto G9 smartphone with the Android app AndroSensor sampled at 100 Hz. All recordings are recorded outside and by the same person. Therefore, the variation in gait and impact from the environment is minimal in our dataset.

a) *Step Counter*: As seen in Fig. 3, at least 300 steps were recorded at walking and running speed with the device positions being in a bag swinging from the arm, the hand, and in a pocket. Steps were also recorded at varying speeds, switching from running to walking with the device in hand and with the device in varying positions and walking speeds.

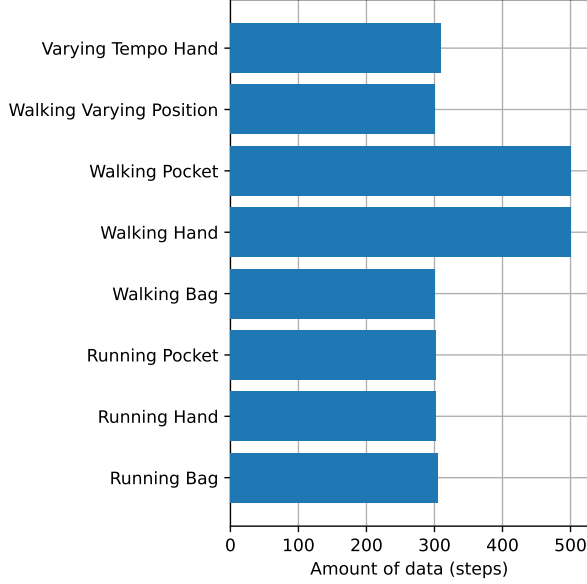


Fig. 3. Amount of steps collected for each phone position.

b) *Activity Recognition*: Running, walking, cycling, sitting, and ascending and descending stairs were recorded with the device in the right front pocket. As seen in Fig. 4 the recorded time for each activity is around 3 hours. The recorded time for ascending and descending stairs deviates from the other activities by a large amount, this is due to the stair activities being significantly harder to record. This discrepancy is alleviated in Section III-C.

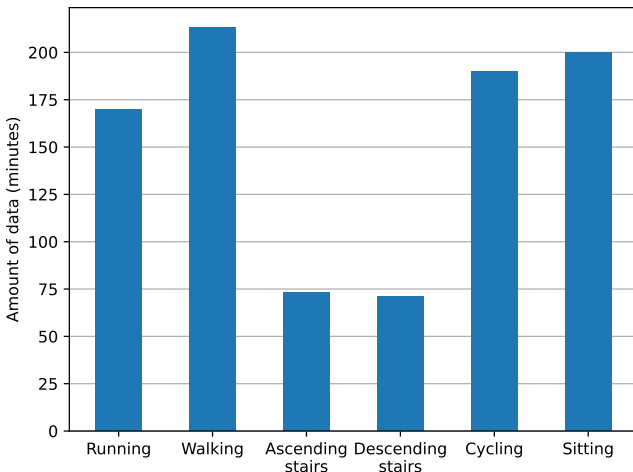


Fig. 4. Amount of data collected for each activity.

2) *Oxford Step Counting Dataset*: The Oxford Step Counting Dataset is a dataset produced by [7] for step counting

algorithm optimization. It contains accelerometer data with an accompanying ground truth count of steps for varying subjects, device positions, and devices. The devices include Google Nexus, Google Pixel, and Samsung smartphones, and the positions are front and back pocket, armband, purse, in hand, and swinging in hand.

3) *Positions and Combined Dataset*: We divide the available data into 4 positions: hand, pocket, swinging, and other. The *hand* position is when the device is in the hand, in front of the subject, such as when the person is texting. While the device is in a swinging motion such as in a bag, purse, or on an armband the arm it's placed in the *swinging* division and when it's placed in the front or back pant pocket it is positioned in the *pocket* division. Other positions such as varying positions, neck pouch, or positions that are undefined in the Oxford dataset are placed in the *other* division.

The division of steps between positions and datasets can be observed in Fig 5. The Oxford dataset is significantly larger than ours, and the data is approximately equally divided between the different positions.

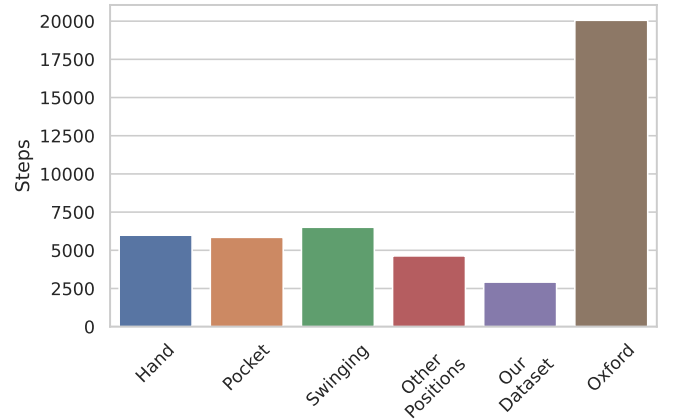


Fig. 5. Amount of steps collected for each position and dataset.

B. Step Counter

The step counter is designed to accurately count steps independent of device, position, and gait. It is implemented using data from the device's accelerometer to find steps. This signal is filtered to remove noise from the sensor as well as small movements not originating from walking, which is then passed through an algorithm to find and validate steps through the use of magnitude and temporal thresholds.

1) *Signal Processing*: The accelerometer is divided into three axes, a_x, a_y, a_z , which are dependent on device orientation. These axes are shown in Fig. 6. To remove the dependency on device orientation, the axes are combined into one signal using the ℓ^2 norm (1)

$$|a|_2 = \sqrt{a_x^2 + a_y^2 + a_z^2}. \quad (1)$$

This results in a noisy signal which is then filtered through a fourth-order BLF with a cutoff frequency of 3 Hz. The resulting signal is approximately sinusoidal with peaks separated by approximately the same time as the time between

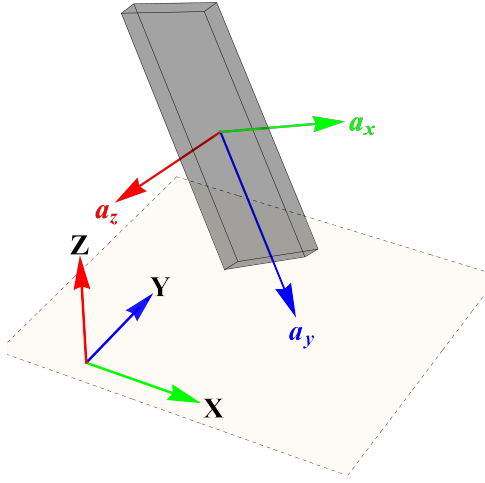


Fig. 6. Acceleration vectors acting on the phone's sensor. The XYZ -axes belong to the Earth and the xyz -axes belong to the phone.

steps. But due to movements not corresponding to walking and the cutoff frequency not being perfectly adjusted to varying walking speeds, not all peaks can be counted as steps.

2) Step Counting and Validation:

Peak Counting: As stated in Section II-A we can use the sinusoidal behavior of the acceleration to count steps. When taking a step the acceleration norm will have maxima and minima, each of these could be used to count the number of steps taken. However, using only one of these is likely to increase the rate of false steps, since maxima or minima can occur from motion unrelated to walking. We will therefore use both maxima and minima, hereafter referred to as peaks and valleys, respectively, to reduce the error rate by a method of step validation.

Magnitude Threshold: In order to count the peaks and estimate steps, a magnitude threshold (threshold variables are hereafter abbreviated Th) was implemented so as to ideally not count peaks not caused by the stepping motion. These peaks had a smaller magnitude than those caused by steps and so were likely to be below the required threshold. In order for a peak to be considered a possible step, the magnitude had to exceed a peak threshold, if it did it was considered a candidate peak. The candidate peak had then to be validated by a valley with a magnitude below one of two valley thresholds discussed in Section III-B2a and III-B2b. In order to account for a possible change in walking speed, the data was divided into 17-second intervals W that contained the samples that were used for the threshold calculations. The peak magnitude threshold Th_p determined the magnitude a peak had to exceed to be considered a candidate peak,

$$Th_p = \mu_W + \frac{\sigma_W}{\beta}, \quad (2)$$

where μ_W is the mean of the data in the interval W , σ_W is the standard deviation, and β is a real constant with a value of 2.25. With this threshold, no peak with a magnitude less than Th_p would be considered a step candidate. If a peak with a magnitude exceeding Th_p was found it had to be validated by a valley with either an absolute or relative magnitude.

a) **Absolute Threshold:** The absolute valley threshold Th_v was defined as

$$Th_v = \mu_W - \frac{\sigma_W}{\beta}. \quad (3)$$

b) **Relative Threshold:** However, it frequently occurred that a valley's magnitude, even though a step was taken, exceeded Th_v and consequently a step was missed. To account for these occurrences the second valley threshold, the relative valley threshold Th_r , was implemented and was related to the magnitude of the peak candidate,

$$Th_r = a_p - \frac{\sigma_W}{\delta}, \quad (4)$$

where a_p is the magnitude of the candidate peak and δ is a real constant with a value of 0.98. Combining (2), (3), and (4) gives the combined peak counting and validating method: a peak exceeding Th_p followed by a valley with a magnitude below Th_v or Th_r counted as a step. Fig. 7-8 presents how a step candidate is at first rejected due to the absolute valley threshold, and then validated using the relative threshold.

c) **Temporal Threshold:** In addition to the magnitude threshold, there was a temporal threshold that limited how separated a valley could be in order to validate a step. This was done to eliminate false steps. During normal stepping motion, a peak-valley pair would follow one another closely, however, if a valley was located for example 10 seconds after a peak, we can be certain that the peak was not caused by a step. In order for the algorithm to handle a change in walking speed within an interval W , a dynamic temporal threshold Th_{pv} was implemented that was proportional to the average time difference t_{pv} between a peak and its validating valley. This threshold came into effect after 2 steps had been counted within the interval so as to not miss the initial steps.

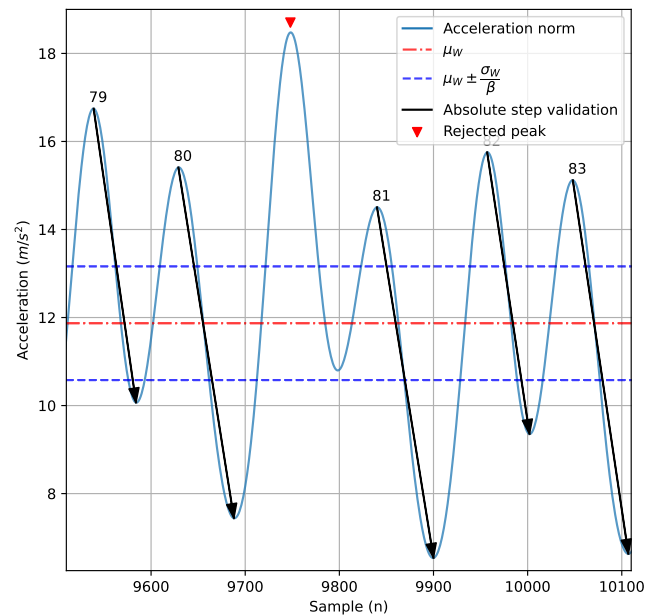


Fig. 7. A peak not counted as a step due to the validating valley being above the absolute threshold Th_v . The number above the peaks indicates the step number.

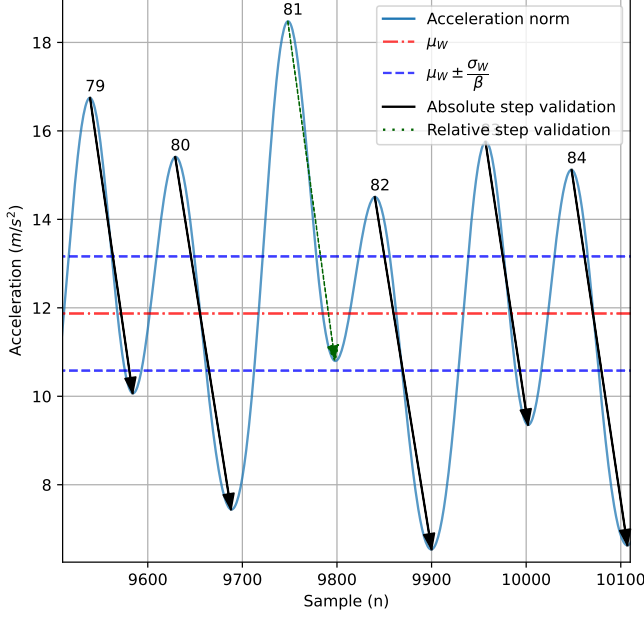


Fig. 8. The rejected peak in Fig. 7 being validated using the relative valley threshold.

$$Th_{pv} = \begin{cases} 10^6 & \text{if } s < 2 \\ 1.1 \cdot (\mu(t_{pv}) + 4\sigma(t_{pv})) & \text{if } s \geq 2 \end{cases}, \quad (5)$$

where s is the number of steps counted within the interval W . If a validating valley that meets the absolute threshold is found that is located beyond the temporal threshold, the algorithm tries to find a valley fulfilling the relative threshold requirement.

The final procedure before a step is fully validated and counted is to determine if between a step candidate P_k and its validating valley V there is another candidate P_{k+1} . If there is, P_k is discarded so that two peaks can not be validated by the same valley, this process is presented in Fig. 9.

The accuracy of the algorithm was tested for each dataset as follows:

$$error = 1 - \frac{\text{counted steps}}{\text{ground truth}} \quad (6)$$

and

$$accuracy = (1 - |error|) \cdot 100\%. \quad (7)$$

Pseudocode for the step counting algorithm can be found in Appendix A.

3) *Optimization of Parameters:* The algorithm uses several parameters such as the length of the interval W , on which the mean μ and the standard deviation σ are calculated, and the threshold parameters β and δ . To determine the optimal configuration of these parameters we minimized the error on the step counting data mentioned in Section III-A1a and III-A2. Through iteration, the most optimal configuration of parameters was found to be: $\beta = 2.25$, $\delta = 0.98$, and the length of the interval W set to 17 seconds.

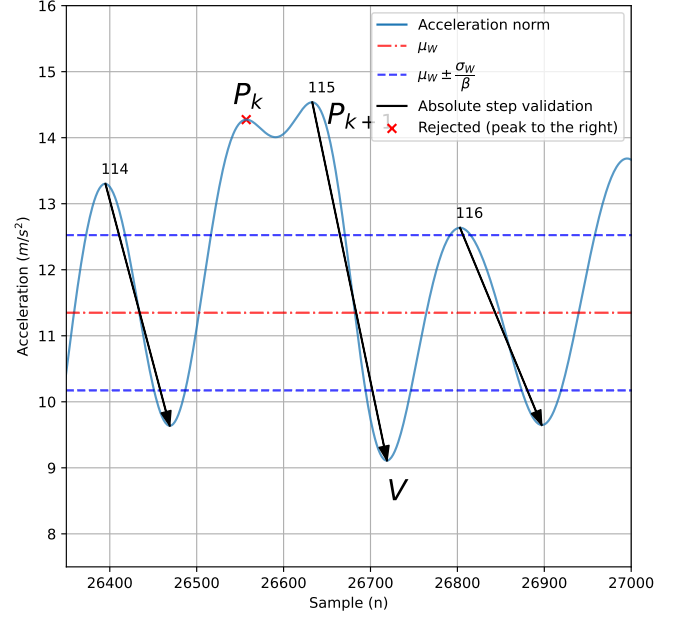


Fig. 9. Peak P_k being rejected due to peak P_{k+1} lying inbetween P_k and the validating valley V .

C. Activity Recognition

The activity recognition model aims to classify several basic activities such as walking, running, sitting still, and ascending or descending stairs. This is done by training several machine learning models on features collected in 10-second intervals from our dataset mentioned in Section III-A1b.

1) *Signal Generation:* Before we extract the features, the signals from which they are extracted have to be generated. These signals are the following:

- Acceleration norm,
- Angular velocity norm,
- Linear acceleration norm,
- Vertical acceleration,
- Heading acceleration,
- Jerk of the acceleration.

The Linear Acceleration (LA), the Vertical Acceleration (VA), and the Heading Acceleration (HA) are generated by rotating the acceleration axes such that the z -axis is parallel to gravity. This rotation is achieved with the help of the Madgwick filter mentioned in Section II-C. The linear acceleration \vec{a}_{LA} is calculated by removing the gravity vector from the acceleration vector \vec{a}_A , which can be done in the rotated system S' as follows:

$$\vec{a}_{LA} = \vec{a}_A - g\vec{e}_{z'}, \quad (8)$$

where g is the gravitational acceleration constant.

The vertical acceleration \vec{a}_{VA} is defined as the z' -axis of the linear acceleration, and the heading acceleration \vec{a}_{HA} is the x' and y' axis of the linear acceleration as follows:

$$\vec{a}_{VA} = (\vec{a}_{LA} \cdot \vec{e}_{z'}) \vec{e}_{z'} \quad (9)$$

$$\vec{a}_{HA} = \vec{a}_{LA} - \vec{a}_{VA} \quad (10)$$

The magnitude of these signals is calculated using the ℓ^2 norm as defined in (1). Finally, the jerk a_J is defined as the first derivative of the acceleration norm.

This signal has been used by other HAR articles such as [15].

As we get the signal in discrete samples we approximate this quantity using linear approximation as follows.

$$a_J = (a[i] - a[i - 1]) \cdot f_s, \quad (11)$$

where f_s is the sample frequency of the signal.

2) *Feature Extraction*: To identify the activity, our model looks at several features from the various signals mentioned in Section III-C1. These features are calculated on 10-second intervals W , and consist of the following properties:

- Mean,
- Standard Deviation,
- Max,
- Min,
- Energy,
- Most Common frequency.

These features are selected to be independent of device orientation and deemed important for activity recognition. Similar features have been used in other projects such as [4] and [5]. Our project differs by using these features on linear, vertical, and heading acceleration.

As mentioned in Section III-A1b the dataset is recorded with a sampling frequency of 100 Hz. These features are therefore calculated on the 1000 samples making up each 10-second interval W .

The features are calculated on each signal s on the intervals W as follows:

Mean:

$$\mu_W = \frac{\sum_{i \in W} s[i]}{1000}. \quad (12)$$

Standard Deviation:

$$\sigma_W = \sqrt{\frac{\sum_{i \in W} (s[i] - \mu_W)^2}{1000}}. \quad (13)$$

Energy: is calculated using Parseval's Theorem

$$E_W = \sum_{i \in W} |s[i]|^2. \quad (14)$$

Most Common Frequency: is defined as the frequency with the maximum amplitude for the Fourier transform of W .

In order to have a more equal amount of data, the features on the stair activities were calculated with 50% overlap, resulting in approximately 150 minutes of stair data. Having unequal amounts of data for each activity increases the risk of the machine learning models being biased towards those activities with more data, reducing the overall accuracy.

Both Neural Networks and SVMs are sensitive to the scale of the data [16], therefore each feature is normalized such that they are scaled to be between 0 and 1. This is done through the function `MinMaxScaler` in the `sklearn` module, which scales each feature v according to

$$v_{normalized} = \frac{v - \min(v)}{\max(v) - \min(v)} \quad (15)$$

3) *Training*: After generating the data points consisting of the 36 resulting features from our dataset. The data points, together with their ground truth activity, are divided into a training and testing dataset with a 2 : 1 division.

The training data is then used to train the RF, ANN, and SVM models. These were implemented in `Python` using the `sklearn` module, and good results were found with 150 estimators and a max depth of 14 for RF, 3 hidden layers containing 50, 50, and 10 nodes respectively using the Adam optimization algorithm for ANN, and SVM using the radial basis function kernel with a regularisation parameter of 8.8 and a kernel coefficient of 2.

These parameters were determined experimentally through iteration by minimizing the error on the testing dataset.

IV. RESULTS

A. Step Counter

The overall accuracy of the step counter is 99.48% on all data in both our dataset and the Oxford dataset combined. These results are shown in Fig. 10, which displays the error of the algorithm at different device positions, on different datasets, and the overall result. It is found that the algorithm is least accurate on swinging motions, where it counts 1.61% fewer steps than the ground truth, but is significantly more accurate while the device is positioned in the pocket with a 0.27% overshoot.

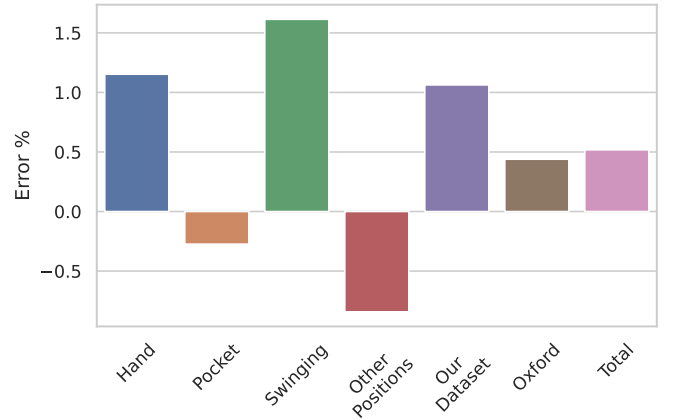


Fig. 10. The error of the step counting algorithm for different device positions, datasets, and overall error. The *pocket* position results in an overshoot of the number of counted steps while the *hand* and *swinging* positions cause undershoot, with the *swinging* position being the least accurate out of all positions. The algorithm is also more accurate on the Oxford dataset than ours, resulting in a total error of 0.52%.

B. Activity Recognition

With the RF, ANN, and SVM machine learning algorithms, the algorithms were 99.7%, 99.6%, and 99.5% accurate in recognizing activities. Fig. 11 presents the confusion matrix for the SVM algorithm, the confusion matrices for RF and ANN models are not presented due to them having any major difference from the SVM matrix. The diagonal squares of the matrix represent where the algorithm correctly recognized the activity. A non-zero value in the other squares means

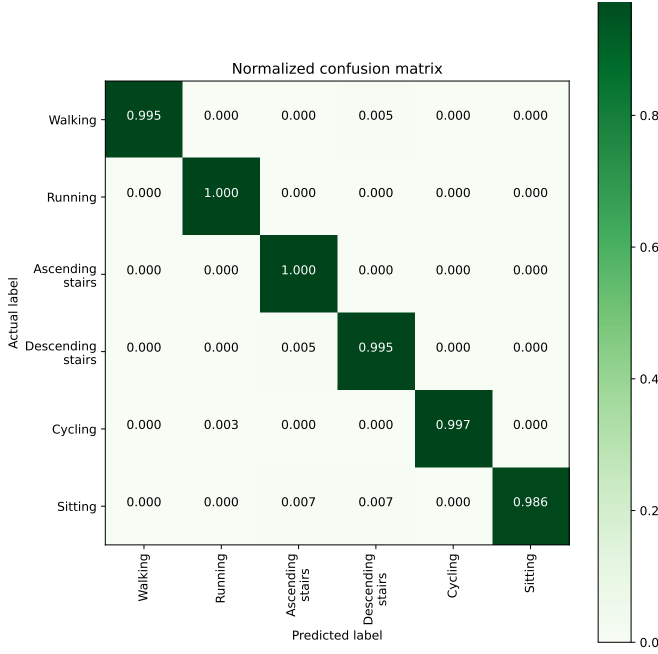


Fig. 11. Confusion matrix for the SVM algorithm.

the algorithm believed the activity to be something other than it was, for example in Fig. 11 it believed “Descending stairs” to be “Ascending stairs” on a few occasions. The RF model provided the highest overall accuracy with a minor improvement over ANN and SVM.

V. DISCUSSION

A. Step Counter

Our results show that the algorithm is off by less than 1% from the ground truth. Although the algorithm is not as accurate as previous step counting algorithms such as [2], it is more accurate than [17] and has proven to be robust. The Oxford data results in Fig. 10 shows that similar results are achieved largely independent of subject, position, and device.

The results also show that our algorithm struggles with swinging motions. This is most likely due to a secondary periodic motion originating from the swing of the arm. This secondary motion can be misinterpreted as steps or the periodic motions can cancel out a peak, such that it does not reach our threshold, such as in Fig. 12.

We may have to be skeptical of the ground truth of the Oxford dataset, after a discussion with the author it has become clear that there are some errors in how they interpreted the ground truth device. There may therefore be a difference between the ground truth reported by the authors of the dataset and the actual number of steps contained in the data, this could either increase or decrease the algorithm’s overall accuracy. This interpretation should have been fixed, but if in a re-evaluation of this algorithm with another dataset other results are achieved, this dataset may come into question.

B. Activity Recognition

The most likely reasons for the high results are homogeneity in the training and testing data, as well as some unexpected

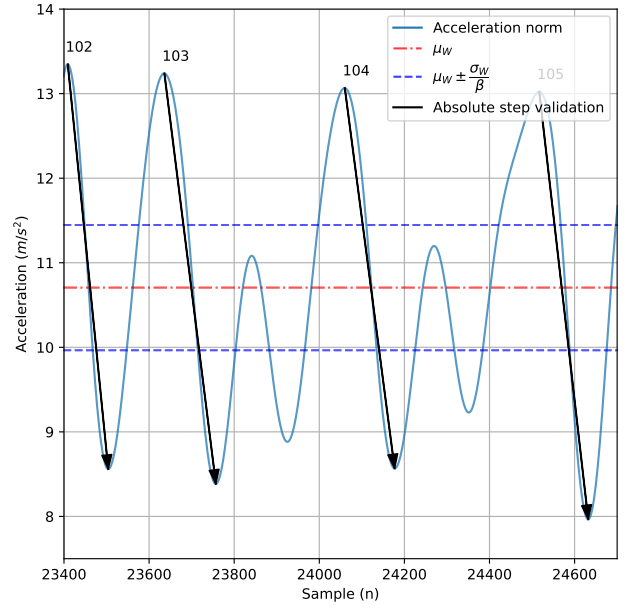


Fig. 12. Two missed steps due to peak-cancellation caused by swinging motions. The acceleration originating from the swinging motion caused enough destructive interference with that of the stepping motion so that two peaks did not reach the threshold.

behaviors in the features. All data used for activity recognition was recorded by the same subject, with the same phone, at the same position, with most recordings being recorded in a few sessions. This leads to homogeneity in the data, which might cause overfitting to a certain environment or style of walking. It is therefore likely that a model trained on this data would not give good results on new subjects or environments.

It is also interesting to look at the impact of features on the model. Using the scikit-learn Python module [16], we can visualize the impact of each feature on the RF model as seen in Fig. 13. Contrary to our expectation that Vertical Acceleration Mean would have a large impact on determining the activity, especially whether the subject is walking up or down stairs, it seems to be negligible. Instead, Heading Acceleration Mean seems to have a larger impact, where we can differentiate the activities almost entirely from this single feature, as seen in Fig. 14. But this behavior is suspicious as stairs seem to be fully separated from walking, even though it is not reasonable that the mean acceleration would triple when walking on plane ground compared to ascending or descending stairs. In the case that walking speed is the same when walking on plane ground as ascending or descending stairs, and if we assume that the mean acceleration is approximately proportional to the velocity, then we can reason that acceleration when walking should not be three times larger than when ascending or descending stairs. This seems to point to the stairs data being recorded at much slower speeds than the walking data, which might unnaturally be observed in reality.

VI. CONCLUSION

This project started with the intention of implementing a functional step counter and HAR model using data from smartphone IMUs. By filtering accelerometer data and counting

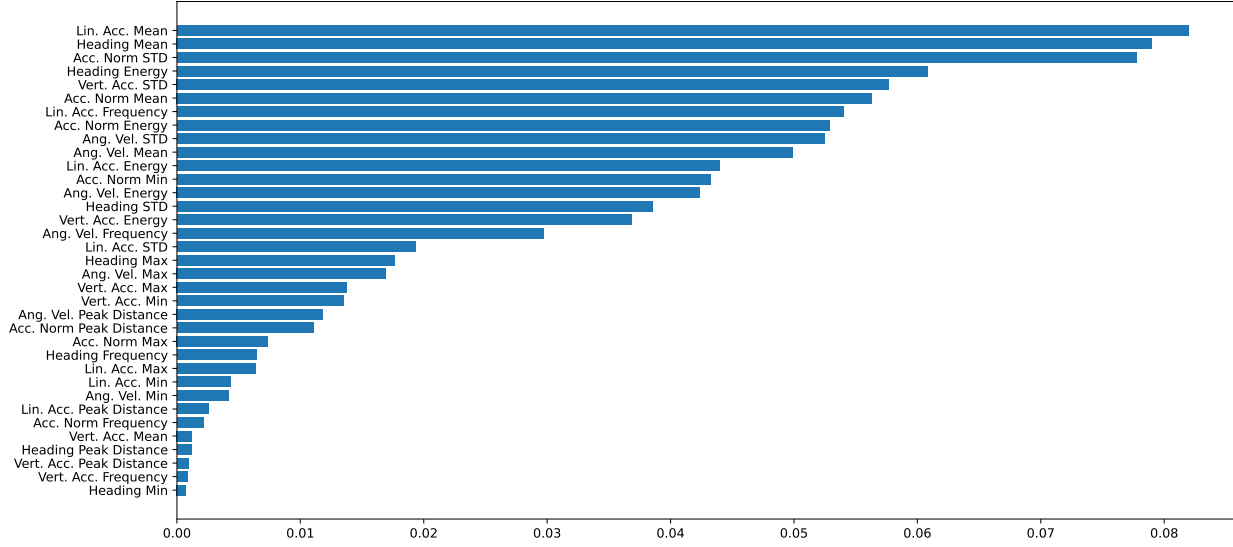


Fig. 13. Feature importance on activity recognition for the RF algorithm, a larger value means the feature has a higher impact on determining activities.

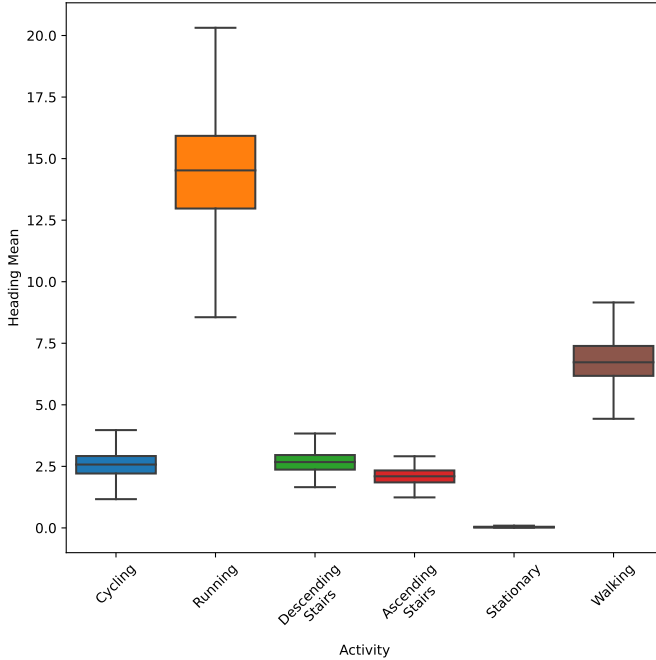


Fig. 14. Distribution of mean heading acceleration between different activities. The figure shows the separation of values between different activities such that Running, Walking, and Stationary can be easily distinguished from each other.

peaks of acceleration with a dynamic threshold, and implementing a method of step validation we've developed a robust step counter that is more than 99.48% accurate on average, with swinging motions being the least accurate at 98.39%. Furthermore, we implemented a method of extracting six different features from the three sensor quantities acceleration, angular

velocity, and linear acceleration and compared the accuracy of the RF, SVM, and ANN machine learning algorithms for HAR. Each of these was greater than 99.5% accurate in their determination of activities, with the RF algorithm providing the highest accuracy of 99.7%. It was also determined that vertical acceleration had an almost negligent impact on activity recognition and heading acceleration had the most impact.

VII. FUTURE WORKS

A. Step Counter

In future works, we would recommend looking into adaptive filters to better preserve signal integrity corresponding to walking and filtering out motion unrelated to walking. Additionally, verification of steps using the gyroscope might help remove or add steps that are discarded or added due to motion unrelated to walking.

B. Activity Recognition

Since the high accuracy of the HAR models is likely due to the homogeneous data, a more diverse dataset could be used to improve generalization and make the model more robust.

VIII. APPENDIX

Appendix A - Pseudo code for the step counter

ACKNOWLEDGMENTS

We would like to thank our supervisor Prakash for his help, feedback, and ideas during the project. We would also like to thank Axel and Björn for their contribution to data collection.

REFERENCES

- [1] WHO. (2022, Apr.) Physical activity. [Fact Sheet]. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/physical-activity>
- [2] M. Khedr and N. El-Sheimy, "A smartphone step counter using imu and magnetometer for navigation and health monitoring applications," *Sensors*, vol. 17, no. 11, Nov. 2017. [Online]. Available: <https://www.mdpi.com/1424-8220/17/11/2573>
- [3] H. Muhsen, O. Al-Amaydeh, and R. Al-Hamlan, "Algorithm design for accurate steps counting based on smartphone sensors for indoor applications," *Advances in Science, Technology and Engineering Systems Journal*, vol. 5, pp. 811–816, Nov. 2020.
- [4] O. Söderberg and O. Blommegård, "Activity recognition using accelerometer and gyroscope data from pocket-worn smartphones," Bsc. thesis, KTH, Stockholm, Sweden, 2021.
- [5] P. Svensson and E. Wendel, "Using machine learning for activity recognition in running exercise," Bsc. thesis, KTH, Stockholm, Sweden, 2021.
- [6] Fiv Asim, "Androsensor," Jan. 2015. [Online]. Available: <https://play.google.com/store/apps/details?id=com.fivasim.androsensor>
- [7] D. Salvi, C. Velardo, J. Brynes, and L. Tarassenko, "An optimised algorithm for accurate steps counting from smart-phone accelerometry," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Jul. 2018, pp. 4423–4427.
- [8] F. Crenna, G. Rossi, and M. Berardengo, "Filtering biomechanical signals in movement analysis," *Sensors*, vol. 21, p. 4580, Jul. 2021.
- [9] W. W. Hoeger, L. Bond, L. Ransdell, J. M. Shimon, and S. Merugu, "One-mile step count at walking and running speeds," *ACSM'S Health & Fitness Journal*, vol. 12, no. 1, p. 14–19, Jan. 2008.
- [10] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of imu and marg orientation using a gradient descent algorithm," in *2011 IEEE International Conference on Rehabilitation Robotics*, Jul. 2011, pp. 1–7.
- [11] (2022, Mar.) Ahrs: Attitude and heading reference systems. [Online]. Available: <https://ahrs.readthedocs.io/en/latest/filters/madgwick.html>
- [12] T. Yiu. (2019, Jun.) Understanding random forest. [Online]. Available: <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>
- [13] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4, no. 4.
- [14] (2022, May) javatpoint. [Online]. Available: <https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm>
- [15] J. Zhu, R. San-Segundo, and J. Pardo, "Feature extraction for robust physical activity recognition," *Human-centric Computing and Information Sciences*, vol. 7, 12 2017.
- [16] A. C. Müller and S. Guido, "Ensembles of decision trees," in *Introduction to machine learning with Python: a guide for data scientists*. Sebastopol, CA: O'Reilly Media, Inc., Oct. 2016, pp. 83–92.
- [17] W. W. Myo, "A more reliable step counter using built-in accelerometer in smartphone," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 12, p. 8, Nov. 2018.

Explaining Mortality Prediction With Logistic Regression

Victor Engdahl and Alva Johansson Staaf

Abstract—Explainability is a key component in building trust for computer calculated predictions when they are applied to areas with influence over individual people. This bachelor thesis project report focuses on the explanation regarding the decision making process of the machine learning method Logistic Regression when predicting mortality. The aim is to present theoretical information about the predictive model as well as an explainable interpretation when applied on the clinical MIMIC-III database. The project found that there was a significant difference between particular features considering the impact of each individual feature on the classification. The feature that showed the greatest impact was the Glasgow Coma Scale value, which could be proven through the fact that a good classifier could be constructed with only that and one other feature. An important conclusion from this study is that a great focus should be enforced early in the implementation process when the features are selected. In this specific case, when medical artificial intelligence is implemented, medical expertise is desired in order to make a good feature selection.

Sammanfattning—Förklarbarhet är en viktig komponent för att skapa förtroende för datorframtagna prognoser när de appliceras på områden som påverkar individuella personer. Denna kandidatexamensarbetsrapport fokuserar på förklarandet av beslutsprocessen hos maskininlärningsmetoden Logistic Regression när dödlighet ska förutsägas. Målet är att presentera information om den förutsäggande modellen samt en förklarbar tolkning av resultaten när modellen appliceras på den kliniska databasen MIMIC-III. Projektet fann att det fanns signifikanta skillnader mellan särskilda egenskaper med hänsyn till den påverkan varje enskild egenskap har på klassificeringen. Den egenskapen som visade ha störst inverkan var Glasgow Coma Scale värdet, vilket kunde visas via det faktum att en god klassificerare kunde konstrueras med endast den och en annan egenskap. En viktig slutsats av denna studie är att stort fokus bör läggas tidigt i implementationsprocessen då egenskaperna väljs. I detta specifika fall, då medicinsk artificiell intelligens implementeras, krävs medicinsk expertis för att göra ett gott egenskapsurval.

Index Terms—Machine Learning, Logistic Regression, Mortality Prediction, Explainability, MIMIC-III.

Supervisors: Ragnar Thobaben

TRITA number: TRITA-EECS-EX-2022:169

I. INTRODUCTION

Every year the concept of Artificial Intelligence (AI) takes one step further away from fiction into our everyday reality. The integration of these algorithms have been so natural that we hardly notice that we use them frequently. The range of applications is vast and include everything from search engines to image recognition. The attractiveness of AI spawns from the ability to create human-like thinking with more effectiveness

and precision than the abilities of an actual human. This is however at the cost of reason. Behind most human decisions there are some sort of reasoning, but since AI has a computer for a brain the decisions are much more binary. Whilst these binary answers are desirable, there are also instances where the reasoning is needed in order to build trust for the algorithms. One example of this is the medical application of AI, which will be the focus point of this bachelor thesis project. Here, the goal is to try to explain how the machine learning (ML) model Logistic Regression (LR) predicts mortality given sets of real patient data. Systems that can alert medical personnel that a patient's vitals might be fatal could be of great use, but only if their predictions can be trusted. A part of this trust could come from providing explainability of the method.

As explained in [1], since the European Union adapted the General Data Protection Regulation (GDPR) in 2016, explainability are in some cases not only a desire to create trust in the algorithms but also a right. People who are subjected to decision making algorithms now have a right to the logic of the verdict. This creates yet another need for explainability and an inclination to deviate from black box algorithms which gives little to no room for explainability.

Whatever the reason behind it, there exists many articles that are trying to give an explanation to ML models. A few examples include [2] which gives a detailed account of the mathematical reasoning behind the classification using LR; and [3] which puts more focus on finding variable significance for mortality prediction to be used in a scoring system to evaluate the risk of heart failure. Yet another example is [4] which was published in the beginning of this year and uses the eICU database to compare the SHapley Additive exPlanations (SHAP) values of four different ML methods to find the importance of different features when predicting mortality. Similarly to this project, [4] found that the Glasgow Coma Scale (GCS) had the biggest impact on the predictions when using a LR model. The difference however, is that in this project there will not only be an interpretation of the prediction results but also an attempt at explaining the mathematical and theoretical aspects of the algorithm.

In this report theoretical information regarding Logistic Regression will be presented in Section II as well as an implementation of such a model on a selection of features from a clinical database in Section III, with the aim to predict the mortality of a patient in an Intensive Care Unit (ICU). With this implementation figures and tables are constructed and presented in Section IV to provide additional interpretation to the model. Lastly the results of these are discussed in Section V to provide a conclusion in Section VI.

II. BACKGROUND AND THEORY

A. Explainability

Due to the ethical concerns associated with health care the main part of this study is explainability. As discussed in [5] the concept of *explainability*, or *interpretability*, is complex and somewhat ill-defined. [5] suggests that the criteria for an explainable ML method can be broadly categorised into transparency or post hoc explanations. In other words, explainability can either focus on precisely how the model works or illustrate why the model has made a certain decision. In an attempt to give as broad of an explanation as possible, this project will focus on both why and how the algorithm had made the choice it has. This permeates the whole project, from the choice of machine learning model to what tests are performed on the classifier.

In this project the explainability is shown by the weight vector that is visualised with different charts and thus explaining which features and vitals are the most important when the model is to predict mortality. Additionally, the importance of the different vitals are further demonstrated via tests where certain vitals are ignored and the classifier acquired then are compared to one when all features are considered.

B. Logistic Regression

As mentioned in the previous section, the choice of machine learning method is dictated by explainability. Thus the choice of method began with some research regarding which machine learning methods are considered explainable. As an example, the study found that methods like deep neural networks, which in later years has become increasingly popular, were not applicable in this project due to the absence of explainability. One method that came up more often than others was logistic regression.

Logistic regression is a machine learning method that builds upon another method called linear regression so in order to understand logistic regression there is first a need to understand linear regression.

1) *Linear Regression*: Let there be one set of vectors that contain some independent features $\mathcal{X} = \{\bar{x}_0, \bar{x}_1, \bar{x}_2, \dots, \bar{x}_n\}$ which all has a corresponding value in $\mathcal{Y} = \{y_0, y_1, y_2, \dots, y_n\}$, it would then be possible to try to make a linear projection $p(\bar{x}_i) = a_0 + a_1x_0 + a_2x_1 + \dots + a_{n+1}x_n$ to y_i . Linear regression works in a way that it assigns random values to $\{a_0, a_1, a_2, \dots, a_{n+1}\}$ and use those values to calculate a prediction $\hat{y}_i = p(\bar{x}_i)$. Since the true value of y_i is already known it is possible to verify if the prediction was accurate or not. Linear regression calculates the error using Mean Squared Error (MSE) over all predictions made from the set of feature vectors, this is called the loss function L .

$$L = \frac{1}{n} \sum ((y - \hat{y})^2) \quad (1)$$

The goal is then to minimise this loss function in order to get the solution that best fits the datasets. This is done with a method called gradient descent, which, depending on the size of the first derivative of the loss functions with respect to all the weights, adjust the weight until the change gets within a

predefined tolerance. When this is accomplished the weights will give the linear projection that best fits the data.

2) *Linear to Logistic Regression*: This linear projection that is acquired from linear regression has no limit, which for a probability is illogical. Also, the regression line we get from linear regression is sensitive towards outliers. This is solved by taking the linear projection from linear regression and using it as a variable to the sigmoid function σ .

$$\sigma(\hat{y}) = \frac{1}{1 + e^{-\hat{y}}} \quad (2)$$

The sigmoid function is bounded to $[0, 1]$ which is appropriate when considering that a probability is calculated. Then a threshold for the probability is chosen and if the probability for a prediction is over the threshold it would be classified as positive, and otherwise negative [6].

C. Mortality Prediction

Since large clinical databases have been made publicly available through cloud storage mortality prediction has become a popular prediction task in ML studies. Due to the simple outcome classification, dead or not dead, it is applicable on many different ML methods which means that there has been a significant number of ways to predict mortality. The difference is not only in the choice of method but also in the choice of features extracted from the databases. Many studies focus on the variables used in various severity classification systems currently used in hospitals, for example APACHE [7] and SAPS [8], which are both scoring systems used in ICUs.

A combination of the variables from these classification systems were used in [9] where the goal was to create an early warning system called EventScore for alerting medical personnel. One of the alert tasks was to alert on the risk of mortality using the Medical Information Mart for Intensive Care (MIMIC)-III data set [10], [11]. The features that were used in [9] are also used in this project and are listed in Table I. These features are taken from each patient and then used as input in the model to predict the outcome of that patient's ICU stay.

TABLE I
FEATURES FOR MORTALITY PREDICTION

Feature	Included in APACHE	Included in SAPS
Diastolic Blood Pressure		
Glasgow Coma Scale	X	X
Glucose		
Heart Rate	X	X
Mean Blood Pressure		
Oxygen Saturation	X	X
Respiratory rate		X
Systolic Blood Pressure	X	
Temperature	X	

As an additional explanation, Glasgow Coma Scale (GCS) is described in [12] as a scale used internationally in clinical assessment of consciousness and measures motor, verbal and eye response. Each of these are evaluated on scales of 1-6, 1-5 and 1-4 respectively, for a total score between 3 and 15

with 15 being the best possible score. In short, the higher on the scale the more conscious the patient is. However, the article mentions that there are some issues with the system, for example the verbal scoring of intubated patients which can lead to a lower scoring in spite of high consciousness, as the verbal scoring depends on the ability to speak.

D. The MIMIC-III Database

This project uses the data from the publicly available clinical database MIMIC-III, which is described in [13] and contains real de-identified patient data from 53,423 different ICU admissions at the Beth Israel Deaconess Medical Center from 2001 to 2012. The database is accessible and downloadable through the creators' website with a required licence from a completed course in Specimen Research. It contains 26 tables that are linked through numerical identifiers (IDs) to keep in line with Health Insurance Portability and Accountability Act (HIPAA) regulations.

According to [13] the numerical IDs serves as a placeholder for the actual identifiers of the patients, such as names and dates. However, time intervals are still important to keep included in the database, to store age for example, and this is solved by shifting all dates with a random offset. As a result all patient admissions occur in the tables between the years 2100 and 2200. To comply with the HIPAA regulations the ages of patients above the age of 89 are also concealed and appear in the tables as being 300 years old.

[13] generalises the contents of these 26 tables as follows; five tables describe and tracks the hospital stay of a patient, five tables are used for cross-referencing codes of diagnoses, procedures and laboratory results and the remaining 16 are filled with information regarding the actual patient care. Patient care includes billing information, measurements, vitals and caregiver observations. All of this data is so called raw data which means that, excluding the de-identification and translation of notes into codes, the data comes straight from the hospital documentation.

III. METHOD

A. Data Pre-processing and Pipeline

To begin with, the data from the MIMIC-III database had to be processed. As described in [14], this means to transform raw real world data into useful and applicable data without changing the core information. The authors in [15] discuss a problem with reproducibility when using publicly available databases since most papers and studies use different preprocessing techniques and highlight the need for a standardised preprocessing framework to be included with these databases. Their solution is a data extraction, preprocessing and representation pipeline they call MIMIC-Extract. This pipeline enables for a simpler way of extracting relevant and useful data that can be immediately implemented in a ML study.

In this report the dataset with default parameters provided by [15] were used. With the pipeline this meant that the 26 original MIMIC-III tables are reduced to only four. Out of these four, only two were needed in this project; the *patients* table and the *vitals_labs_means* table. These tables that are

produced with the pipeline have corrected outliers, meaning that obscure and unreasonable values have been changed. This is done by replacing the values of the mild outliers with the nearest valid values and removing patients with values that are deemed extreme outliers. The processing also includes defining a cohort of patients which by default means that the only patients included are those over the age of 15. Furthermore, only their first ICU stay during a hospital admission that was above 12 hours and less than 10 day were included. This gives a remainder of 34,472 patients in comparison to the original 53,423 that are included in the MIMIC-III database.

B. Feature Selection

The features extracted from the processed MIMIC-III database were those used in [9] and listed in Table I in addition to age, gender, ethnicity, length of each ICU stay and whether or not the patient died in the ICU. Age and length of stay were regulated by the default cohort generated by the pipeline and the vital features were regulated by the outlier detection and elimination, both described in section III-A. All extracted features as well as their value ranges are included in Table II and were collected for all 34,472 ICU stays.

TABLE II
FEATURES, VALUE RANGE, UNITS WHEN APPLICABLE AND MISSING PERCENT

Feature	Value	Unit	Missing %
Gender	Female/Male	-	0
Age	15-300 *	years	0
Ethnicity	**	-	0
Diastolic Blood Pressure	13.40-127.65	mmHg	0.58
Glasgow Coma Scale	3-15	-	43.37
Glucose	44-640	mg/dL	0.29
Heart Rate	30.25-146.57	bpm	0.59
Mean Blood Pressure	31.17-140.65	mmHg	0.59
Oxygen Saturation	22.7-100.0	%	0.29
Respiratory rate	6.44-48.0	breaths/minute	0.66
Systolic Blood Pressure	15.57-197.77	mmHg	0.59
Temperature	30.58-39.72	°C	0.84
Length of Stay	12-239	hours	0
Dead in ICU	0/1	-	0

* Ages over 89 are recorded as 300

** Ethnicity: 40 different values were included

However, when studying the data it became apparent that some ICU stays did not contain all the features and that some features seemed to be missing more often than others. This in spite of the claim of low missingness from [15]. Among the extracted features there were some that were missing from patients more frequently, most notably the Glasgow Coma Scale which were missing from 43.37% of the ICU stays.

As previously stated the tables used from the pipeline were the *patients* and *vitals_labs_means* tables. From the first table the static features; gender, age and ethnicity, as well as the features regarding each stay; length of stay and dead in ICU were taken. The rest of the features were taken from the second

table which consists of hourly means of each vital feature. Since the goal of the project was explainability of a model it was opted to take the mean of all hourly values associated with each ICU stay rather than looking at each hour individually. To deal with the missingness, the patients which had a missing feature value were removed from the data set which reduced the number of patients in the fully processed data set to 19,410 individual ICU stays. Of these, 8,325 were females, 11,085 were males and in total 1,331 had a recorded fatality in the ICU.

	predicted negative	predicted positive
actual negative	TN	FP
actual positive	FN	TP

Fig. 1. Example of a confusion matrix.

C. Logistic Regression Implementation

For all of the machine learning tasks in this project the python library *scikit-learn* from [16] was used. The class *Logistic Regression* is documented in [17] and implements the ML method logistic regression (LR) and has 15 parameters shown in Table III. The same documentation shows that the class also has ten methods.

Using the *scikit-learn* library and the method *test_train_split*, which is documented in [17], the 19,410 ICU stays were split into four sets; *x_train*, *x_test*, *y_train* and *y_test*. The *x* sets were the feature vectors and the *y* sets the corresponding classifier of dead or not dead and the suffix of *train* or *test* signifies if the set will be used for the training or the testing of the LR model. When using this method two of the parameters were adjusted from the default value, *test_size* and *random_state*, which were set to 0.2 and 0 respectively. This means that the size of the *test* set are 20% of the complete data set which according to [18] is within the common range of 20-40%. The *random_state* parameter ensures that the sets are the same every time the script is ran to enable for better interpretation of the results.

With this split the method of the *Logistic Regression* class named *fit* could be used to fit the model to the training data sets. The method *score* could then be used to give a score of the model in regards to the *test* set. With this score an evaluation of the hyperparameters of the model could be performed which gave the parameter values listed in Table III. It should be noted however, that these values do not stray far from the default values listed in [17].

TABLE III
SKLEARN LOGISTIC REGRESSION PARAMETERS FROM [17]

Parameter	Description	Used value (D) indicates default
penalty	Specifies the norm in penalization	l1
dual	Only used for l2 penalization	False (D)
tol	Stopping criteria tolerance	1e-4 (D)
C	Regularization strength inverse	1.623776739188721
fit_intercept	If True, adds a constant to the decision function	True (D)
intercept_scaling	A constant with this value is appended to the instance vector	1 (D)
class_weight	Decides class weight. If None, all class weight equal to one	None (D)
random_state	Seed for random number generator which is used to shuffle the data	None (D)
solver	Declares algorithm for optimization problem	liblinear
max_iter	Only for newton-cg, sag and lbfgs solvers	100 (D)
multi_class	Only for newton-cg, sag and lbfgs solvers	auto (D)

D. Visualisation

The performance of an algorithm can be measured in a variety of ways. With the goal of explainability it was opted to use three different plots to visualise how well the model achieved an accurate classification under different circumstances and they were as follows:

- 1) A confusion matrix (CM), which according to [19] illustrates how many of the data points are classified as true positive (TP), false positive (FP), true negative (TN) and false negative (FN). An example of a confusion matrix can be found in Fig. 1. The same article shows how the values of the confusion matrix can be used to calculate the recall and precision values of a model, as well as the true and false positive rates, which are defined as

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

$$TruePositiveRate(TPR) = \frac{TP}{TP + FN} \quad (5)$$

$$FalsePositiveRate(FPR) = \frac{FP}{FP + TN} \quad (6)$$

which all are used as another measurement of the performance.

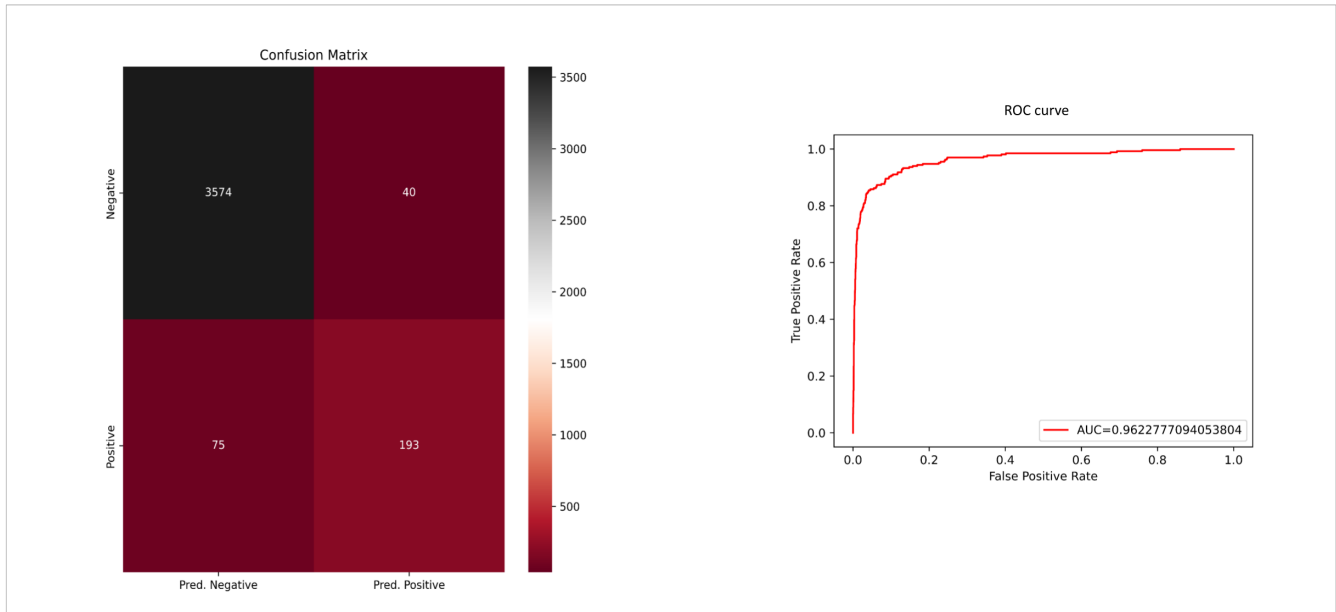


Fig. 2. Confusion Matrix and ROC curve evaluations of the LR model with threshold 0.63.

- 2) A Receiver Operator Characteristic (ROC) curve as well as the Area Under the Curve (AUC) which, according to [18], is used to compare performances of models. The ROC curve plots the TPR against the FPR, as described by Equations (5) and (6), for different threshold values. A good ML model produces a ROC curve which follows the top left corner of the graph as closely as possible which in turn means that the AUC is close to one.
- 3) A bar chart visualising the value of the coefficients in respect to each other, and thus showing the importance of a feature.

IV. RESULTS

A. Threshold

The threshold of the logistic regression model was adjusted to give best possible performance measured with the CM and AUC plots. There needed to be a deviation from the standard threshold value of 0.5, since this value made the classification of positive data points difficult. A higher threshold leads in this implementation to a greater amount of positive classifications. Here a positive data point, or patient, is a deceased patient since the aim is to predict mortality, where as a negative patient is the opposite. With the 0.5 threshold the model had problems with correctly classifying positive patients and had a recall of 0.65 which meant that many of the positives were falsely classified as negatives. The combined recall and precision values were found to be the greatest at a threshold of 0.63. This gave a model according to Fig. 2 which instead had a recall of 0.72 and an AUC close to 1.

B. Patient Values

Listed in Table IV are the median values of the groups of predicted positive and negative patients in regards to all twelve features. This is done to showcase which values differ between

the two groups as to better understand which features have an impact on the classification. The medians are taken each feature individually which means that the values collectively does not deliberately correspond to any patient in the data set. The median values are similar in both groups in regard to most of the features but with a bigger difference in the GCS, glucose, heart rate and systolic blood pressure. Out of these, all but the first have a sizable distribution of the values in the total data set, as shown in Table II.

TABLE IV
MEDIAN OF PREDICTED POSITIVES AND PREDICTED NEGATIVES

Feature	Median Values of Predicted Positives	Median Values of Predicted Negatives
Gender	M	M
Age	68.96	65.54
Ethnicity	WHITE	WHITE
Diastolic Blood Pressure	55.89	59.17
Glasgow Coma Scale	5.81	14.33
Glucose	160.68	127.83
Heart Rate	94.48	83.30
Mean Blood Pressure	73.01	78.57
Oxygen Saturation	95.78	97.10
Respiratory rate	19.27	18.40
Systolic Blood Pressure	105.63	119.61
Temperature	37.01	36.93

C. Variable Coefficients

The median values show an indication of some features having more of an impact on the classification than others. This could better be illustrated through the coefficients, or weights, of each feature which can be found in Fig. 3. A large absolute

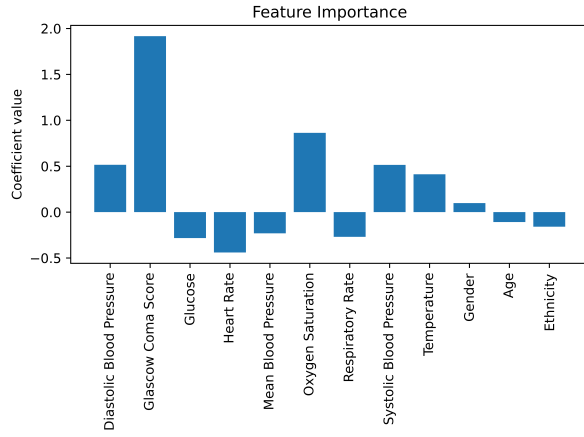


Fig. 3. The coefficient value of each of the features.

value show a high feature significance in the construction of the model as that feature is weighed more when the probability is calculated, as described in section II-B. The biggest coefficients are those of the GCS and oxygen saturation which have absolute values of 1.92 and 0.86 respectively. An additional comment relevant to Fig. 3 is that a positive coefficient value increases the probability of survival, whilst a negative value decreases it.

D. Changing Features

Table V illustrates what happens to the model if each of the features are removed from the model implementation, one at a time. The goal of this is to show which features contribute to a good model performance by highlighting the difference in the AUC. The biggest difference between the original AUC of 0.96228 and the model performance when one feature is removed is when the GCS value is removed. This results in an AUC of 0.78446 which is a relatively big difference in comparison to the rest of the values. Since both Table V and Fig. 3 implicate that GCS has a substantial influence over the classification of the patients, the same process of removing one feature at a time was performed with the GCS feature already removed. This is also shown in Table V.

E. Simpler Model

Fig. 7 shows the CM and the AUC of the model with only GCS and oxygen saturation as features as these two were the ones shown to have the most impact on the classification. This was done to see if these two were enough to create an equivalent model to the one using all twelve features. The AUC of the one with two features was 0.93254 which is quite close to original AUC of 0.96228.

The importance of GCS is also noticeable when looking at Fig. 4 which shows the calculated probability of each patient and the GCS for each of these patients as well as the classification made by the model. The threshold of 0.63 is also plotted. The data points classified as positive reside closer to the bottom left corner of the plot where as the negatives are at the top right. If a function curve were to be fitted to the

TABLE V
AUC WHEN REMOVING ONE FEATURE AT A TIME AS WELL AS WHEN GLASGOW COMA SCALE WAS ALREADY REMOVED. AUC CLOSER TO 1 INDICATES BETTER PERFORMANCE.

Feature Removed	AUC	AUC with GCS removed
Gender	0.96163	0.78379
Age	0.96212	0.77986
Ethnicity	0.96269	0.78484
Diastolic Blood Pressure	0.96118	0.77714
Glasgow Coma Scale	0.78446	-
Glucose	0.96015	0.75825
Heart Rate	0.96202	0.76911
Mean Blood Pressure	0.96201	0.77920
Oxygen Saturation	0.95239	0.76665
Respiratory rate	0.95938	0.78592
Systolic Blood Pressure	0.96112	0.77327
Temperature	0.95537	0.78504

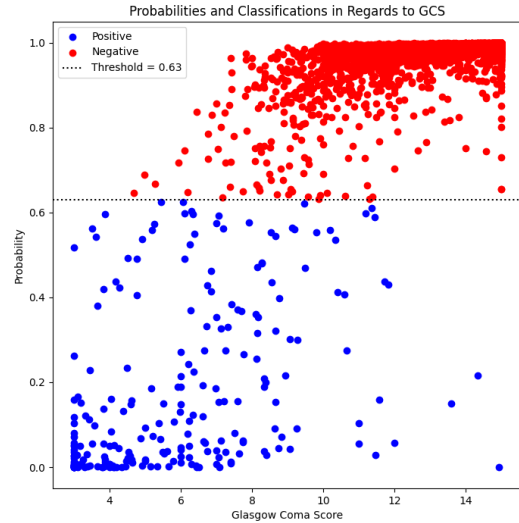


Fig. 4. Probability vs GCS from the model with all twelve features with the predicted classification of each data point.

plotted data points the curve would resemble the recognizable S shape of the sigmoid function curve, which shows that the model adapts well to the data with regards to GCS. This is due to the fact that a higher GCS generally leads to a higher probability which means a negative classification. There are three data points that clearly do not fit into the S shape when only looking at GCS. These are the data points with a GCS of 13 or above and a probability of maximum 0.25. The fact that they do not seem to fit means that there are other feature values than their GCS that contribute to their classification. All feature values of these three patients are listed in Table VI.

This shows that Patient 1 has values that all differ from the means of both predicted positives and negatives, Patient 2 has an oxygen saturation lower than the mean of both

classifications and Patient 3 has a significantly higher glucose value. As shown in Fig. 5 both Patient 2 and Patient 3 did actually die in the ICU where as Patient 1 did not and is therefore a false positive. This figure also illustrates the imperfections of the predictions of the model. Even though there are significant clusters of red and blue data points in the upper right and lower left corners respectively, there are a noticeable number of "wrongfully" colored data points within these clusters as well. The dispersion of the positive blue data points can be correlated to the high number of false negatives as shown in Fig. 2 since all of the positive data points above the threshold line are false negatives.

TABLE VI
FEATURE VALUES OF DEVIATING DATA POINTS

Feature	Patient 1	Patient 2	Patient 3
Gender	M	M	M
Age	300	36.89	300
Ethnicity	WHITE	OTHER	WHITE
Diastolic Blood Pressure	47.95	60.12	58.07
Glasgow Coma Scale	13.6	14.93	14.33
Glucose	155.0	112.57	197.0
Heart Rate	118.42	93.72	117.30
Mean Blood Pressure	68.68	75.40	68.20
Oxygen Saturation	89.61	52.05	91.20
Respiratory rate	30.66	24.13	29.14
Systolic Blood Pressure	110.16	106.41	92.79
Temperature	36.40	36.89	35.80

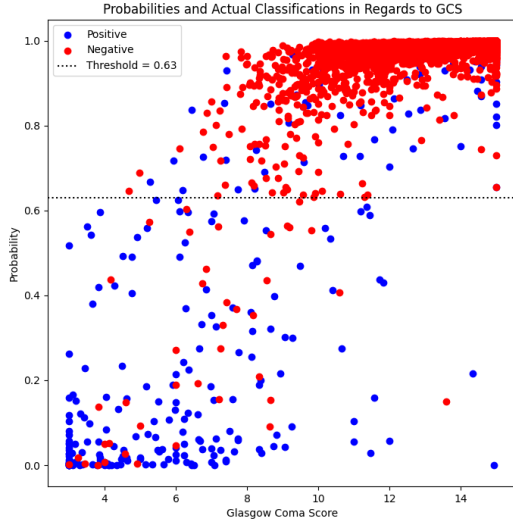


Fig. 5. Probability vs GCS from the model with all twelve features with the actual classification of each data point.

F. Random Forest

To compare the results from the LR model a simple implementation of the ML method Random Forest (RF), also from the *scikit-learn* library [17], was used on the same features

as in the original LR implementation. The application of this RF model gave the coefficient values as shown in Fig. 6. It is notable that unlike with LR, RF only give positive coefficient values. The three most significant features, those with the highest coefficients, were GCS, oxygen saturation and systolic blood pressure.

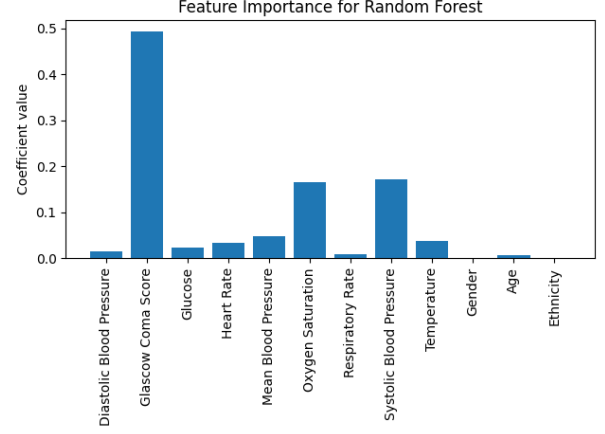


Fig. 6. The coefficient value of each of the features when using the RF model.

V. DISCUSSION

A. Threshold Correlation

The adjustment of the threshold was a result of the imbalanced total data set. Only 7% of the total ICU stays were positive data points. Because of this imbalance the recall value became the significant when looking at the model performance at different threshold as this value is linked to the rate of positive classifications. Another reason for the importance of the recall value was the fact that the aim was mortality prediction, with a long term goal of actually creating a useful prediction model. In this case a false negative is much more harmful than a false positive. The increase in threshold meant a sacrifice in negative classifications in favour of positive classifications but there still had to be a reasonable middle ground between the two as an increase means a larger amount of false positives which leads to a decreased precision. This is why the use of AUC was motivated when comparing model performance in this case.

B. Most Important Feature

The project also found that the most important feature in predicting mortality out of the twelve features used was Glasgow Coma Scale. This is a reasonable result when looking at what GCS is actually used for, to evaluate consciousness. It is also notable that this project uses the mean of all feature values thorough out the entire stay, which Table II showed ranged from 12 to 239 hours depending on the patient. This means that a patient which had a stay of 239 hours, or about 10 days, could have a low value when entering the ICU and have a higher value when leaving, altering the mean value. The reason for using the mean in spite of this was that there

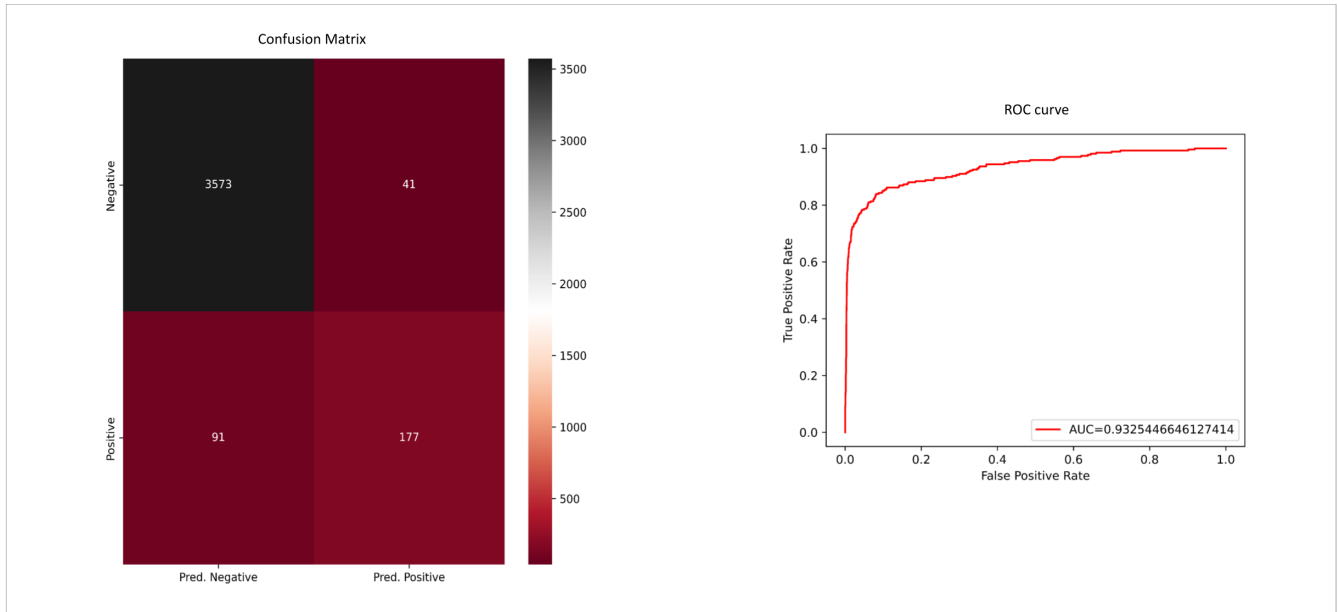


Fig. 7. Confusion Matrix and ROC curve evaluations of the LR model with only GCS and oxygen saturation as features.

were large irregularities in how often and how many scorings that had been done for each ICU stay.

As previously noted, a high GSC leads to a high probability of survival, however there are some outliers to this generalisation that still are correctly classified. This highlights the fact that even if GCS is the most important feature, the other features contribute to the correct classification of data points where GCS is insufficient. In Table VI three outliers are listed with their values in order to see what contributed to their classification when GCS indicated survival, and interestingly two out of these three were classified correctly.

C. Verification With Random Forrest

A correlation between the classification and the GCS value can also be found in the Random Forrest implementation which further supports the importance of the GCS value of each patient. Even though this model is not explained in any detail, as the Logistic Regression model, it is clear that the GCS has more importance than any of the other features, as illustrated by Fig. 6. The other features that clearly have a bigger impact on that model can also be found to be amongst those with larger coefficient values in the LR model.

D. Two Feature Model

The performance of the model using only GCS and oxygen saturation as features was found to be fairly similar to the model using all twelve features. The AUC of the two were similar but the ROC curve looked different. This can be explained by the confusion matrix of each model as the original had 75 false negatives and the two feature model had 91 but they had 40 and 41 false positives respectively. This relates to TPRs and FPRs of 0.7191 and 0.0111 for the original and 0.6604 and 0.0113 for the two feature model. This indicates that feature selection is a crucial part of the

implementation of medical AI, and highlights the importance of medical expertise in the area. With this in mind it is not unlikely that an even better classifier could be achieved, not only with more data but with features that have a bigger significance. Less features that contribute more would also decrease the workload that is needed to train the classifier.

VI. CONCLUSION

During the research in this bachelor thesis project it has been established that the most important feature in this implementation is GCS which seems reasonable since it is a measure of consciousness. At the same time, it has been clear that it is important to consider multiple features to get a well rounded classifier. However, the result of the project seem to indicate that it is preferable to have fewer but more significant features. This highlights the importance of the earlier steps of AI implementation, feature selection, and in the particular case of medical AI, the importance of medical expertise.

VII. FUTURE WORK

The goal of this project was to give explainability to a machine learning method which is why the usefulness of the method was not a priority. To develop a logistic regression model with real application possibilities a suggestion would be to look at hourly data instead of the mean of the entire ICU stay. This would enable for a warning system where as the model used in this project only works in hindsight, as all of the patient data needs to be available. It would be more applicable to develop a system which could predict the mortality probability given initial data points. However, there is reason to believe that the features found to be the most deciding in this project are the ones to look at when implementing such a warning system.

ACKNOWLEDGMENT

The group would like to thank their supervisor Ragnar Thoben, for his support, guidance and patience throughout the whole project. A special thanks would also like to be extended to Anna Sundelin for taking her time to explain and help decipher the medical aspects of this project.

REFERENCES

- [1] B. Goodman and S. Flaxman, “European union regulations on algorithmic decision-making and a “right to explanation,”” *AI magazine*, vol. 38, no. 3, pp. 50–57, Aug. 2017.
- [2] S. Sperandei, “Understanding logistic regression analysis,” *Biochemia medica*, vol. 24, no. 1, pp. 12–18, Feb. 2014.
- [3] D. S. Lee, P. C. Austin, J. L. Rouleau, P. P. Liu, D. Naimark, and J. V. Tu, “Predicting mortality among patients hospitalized for heart failure: derivation and validation of a clinical model,” *Jama*, vol. 290, no. 19, pp. 2581–2587, Nov. 2003.
- [4] E. Stenwig, G. Salvi, P. S. Rossi, and N. K. Skjærvold, “Comparative analysis of explainable machine learning prediction models for hospital mortality,” *BMC Medical Research Methodology*, vol. 22, no. 1, pp. 1–14, Feb. 2022.
- [5] Z. C. Lipton, “The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery,” *Queue*, vol. 16, no. 3, pp. 31–57, Jun. 2018.
- [6] sourav6458. (2020, Dec.) Linear vs logistic regression: Linear and logistic regression. [Online]. Available: <https://www.analyticsvidhya.com/blog/2020/12/beginners-take-how-logistic-regression-is-related-to-linear-regression/>
- [7] C. Berry. (2020, Nov.) Critical care scoring systems. [Online]. Available: <https://www.msmanuals.com/professional/critical-care-medicine/approach-to-the-critically-ill-patient/critical-care-scoring-systems>
- [8] R. P. Moreno, P. G. H. Metnitz, E. Almeida, B. Jordan, P. Bauer, R. A. Campos, G. Iapichino, D. Edbrooke, M. Capuzzo, and J.-R. Le Gall, “Saps 3—from evaluation of the patient to evaluation of the intensive care unit. part 2: Development of a prognostic model for hospital mortality at icu admission,” *Intensive Care Medicine*, vol. 31, no. 10, pp. 1345–1355, Aug. 2005.
- [9] I. Hammoud, P. Prasanna, I. Ramakrishnan, A. Singer, M. Henry, and H. Thode, “Eventscore: An automated real-time early warning score for clinical events,” *arXiv preprint arXiv:2102.05958*, Feb. 2021.
- [10] A. Johnson, T. Pollard, and R. Mark. (2016, Sep.) MIMIC-III clinical database (version 1.4). [Online]. Available: <https://doi.org/10.13026/C2XW26>
- [11] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, “Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals,” *circulation*, vol. 101, no. 23, pp. e215–e220, Jun. 2000.
- [12] G. L. Sternbach, “The glasgow coma scale,” *The Journal of emergency medicine*, vol. 19, no. 1, pp. 67–71, Feb. 2000.
- [13] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, “MIMIC-III, a freely accessible critical care database,” *Scientific data*, vol. 3, no. 1, pp. 1–9, May 2016.
- [14] “Data preprocessing and intelligent data analysis,” *Intelligent Data Analysis*, vol. 1, no. 1, pp. 3–23, Jan. 1997.
- [15] S. Wang, M. B. McDermott, G. Chauhan, M. Ghassemi, M. C. Hughes, and T. Naumann, “Mimic-extract: A data extraction, preprocessing, and representation pipeline for mimic-iii,” in *Proceedings of the ACM conference on health, inference, and learning*, Toronto, Canada, Apr. 2020, pp. 222–235.
- [16] 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.
- [17] scikit-learn developers. (2017, Jun) scikit-learn user guide. [Online]. Available: https://scikit-learn.org/0.18/_downloads/scikit-learn-docs.pdf
- [18] L. A. Celi, M. S. Majumder, P. Ordóñez, J. S. Osorio, K. E. Paik, and M. Somai, *Leveraging data science for global health*. Cham, Switzerland: Springer Nature, 2020.
- [19] J. Davis and M. Goadrich, “The relationship between precision-recall and roc curves,” in *Proceedings of the 23rd international conference on Machine learning*, Pittsburgh, PA, Jun. 2006, pp. 233–240.

Mortality Prediction in Intensive Care Units by Utilizing the MIMIC-IV Clinical Database

Raymond Wang

Abstract—Machine learning has the potential of significantly improving daily operations in health care institutions but many persistent barriers are to be faced in order to ensure its wider acceptance. Among such obstacles are the accuracy and reliability. For a decision support system to be entrusted by the medical staff in clinical situations, it must perform with an accuracy comparable to or surpassing that of human medics, as well as having a universal applicability and not be subject to any bias. In this paper the MIMIC-IV Clinical Database will be utilized in order to: (1) Predict patient mortality and its associated risk factors in intensive care units (ICU) and: (2) Assess the reliability of utilizing the database as a basis for a clinical decision system. The cohort consisted of 523,740 hospitalizations, matched with each respective admitting diagnoses in ICD-9 format. The diagnoses were then converted from code to text-format, with the most frequently occurring factors (words) observed in deceased and surviving patients being analyzed with an Natural language Processing (NLP) algorithm. The results concluded that many of the observed risk factors were self-evident while others required further explanation, and that the performance was highly by selection of hyperparameters. Finally, the MIMIC-IV database can serve as a stable foundation for a clinical decision system but its reliability and universality shall also be taken into consideration.

Sammanfattning—Maskininlärningstekniker har en stor potential att gynna sjukvården men står inför ett flertal hinder för att fullständigt kunna tillämpas. Framförallt bör modellernas tolkningsbarhet och reproducerbarhet beaktas. För att ett kliniskt beslutstödssystem skall vara fullständigt anförtroddt av sjukvårdspersonal måste det kunna prestera med en jämförbar eller högre träffsäkerhet än sjukvårdspersonal, samt kunna tillämpas i åtskilliga sammanhang utan någon subjektivitet. Syftet med denna studie är att: (1) Förutspå patientdödsfall i intensivvårdsavdelningar och utreda dess riskfaktorer genom journalförd information från databasen MIMIC-IV och: (2) Bedöma databasens tillförlitlighet som underlag för ett kliniskt beslutstödssystem. Kohorten bestod av 523,740 hospitaliseringar som matchades med de diagnoser som ställdes vid deras sjukhusintag. Eftersom diagnoserna inskrevs i ICD-9-format omvandlades dessa till ord och de mest förekommande faktorerna (orden) för avlidna och överlevande patienter analyserades med en NLP-modell (Natural Language Processing). Resultaten konkluderade att många av de förutspådda riskfaktorerna var uppenbara medan andra krävde ytterligare klargöranden. Dessutom kunde val av hyperparametrar stort påverka modellens kvalitet. MIMIC-IV-databasen kan utgöra ett gediget underlag för ett kliniskt beslutssystem men dess tillförlitlighet och relevans bör även tas i beaktande.

Index Terms—Clinical Data Science, Mortality Prediction, MIMIC-IV, Machine Learning, NLP, ICU

Supervisor: Ragnar Thobaben

TRITA number: TRITA-EECS-EX-2022:170

I. INTRODUCTION

In a daily basis, an immense amount data and especially electronic health records (EHR), are generated in hospital facilities. However, they will most likely be stored across different locations and departments resulting in so-called data-fragmentation [1]. EHR databases may also be complex and difficult to use, with a myriad of actions to take in order to extract the clinically relevant information. This may pose a great difficulty when applying machine learning techniques to analyze such kind of data, which has the potential of being able to significantly improve daily operations in healthcare [2]. In order to realize this potential and address its associated difficulties, the MIMIC-IV (Medical Information Mart for Intensive Care) Clinical Database will be utilized, which provides de-identified records of patients admitted at the Beth Israel Deaconess Medical Center in Boston from 2008-2019. MIMIC IV is an update to the previous MIMIC-III Clinical Database, which adopts a more "modular approach to data organization" [3]. The MIMIC Clinical Database has been shown to be popular in various clinical data science applications [4]. However, since the majority of researched articles do not share their codes, only redundant efforts have been made build upon existing pipelines resulting in difficulties elaborating upon possible dissimilarities in results. MIMIC-IV rectifies the complications of data-fragmentation of EHR records by storing data from different hospital departments in the same database, and although several improvements have been made from its predecessors, it has still been proven to be difficult to use [5].

This paper aims to predict the probability of mortality for patients in-hospital deaths by utilizing the MIMIC-IV Clinical Database. This in order to:

- 1) Elaborate upon risk factors most commonly associated with patient mortality in ICU settings.
- 2) Assess the reliability of utilizing MIMIC-IV data as a basis for a hospital decision support system.

Mortality prediction has shown to be an often-occurring topic in many papers about the MIMIC Clinical Database [5]. However, few have used the recent MIMIC-IV update in such tasks and it is therefore meaningful to assess the clinical applicability of MIMIC-IV and possibly, in comparison to previous versions.

Mortality prediction has the potential of yielding early identification of high-risk patients and making room for improved treatment [6]. The premise of such kinds of prediction algorithms relies upon the records of patient diagnoses recorded by physicians during hospital admission.

These as well as death dates for non-surviving patient are all included in the MIMIC-IV database. However MIMIC-IV lists patient diagnoses in ICD 9-format [3] while previous versions such as MIMIC-III also contain physician notes for each patient admission, [7]. In this instance, the ICD 9-code will be converted to the specific diagnosis in text-format utilizing a conversion table provided by the MIMIC-IV database. The extracted words allow for a creation of a singular data set, setting stone for the implementation of a mortality prediction algorithm utilizing Natural Language Processing (NLP).

II. BACKGROUND

A. Machine Learning

Due to the ubiquity of Artificial Intelligence and Machine Learning (ML) in our daily lives, many have termed it as a quintessential facet of the fourth industrial revolution [8]. The premise of machine learning algorithms is to enable computer software to autonomously identify patterns from sample data, in order to make predictions or decisions. This can be applied to many fields of areas and as such, different types of algorithms and methods are utilized for different types of problems [9]. While this study utilizes an NLP algorithm, other algorithms such Random Forests and Convolutional Neural Networks (CNN) are also commonly implemented in machine learning problems. The latter of which is ubiquitous in the field of image classification [10].

B. Classification methods

Machine learning problems fall into three general categories: (1) learning (2) Unsupervised learning and (3) Reinforcement learning. Supervised learning refers to an algorithm that has been trained with a labeled data set and making predictions of specified terms. Such algorithms map inputs x to outputs y given a data set $D = \{x_i, y_i\}_{i=1}^N$ containing N numbers of data points [11]. On the contrary, unsupervised learning refers to an algorithm trained with trial and error in order to identify patterns in unlabeled data [12]. Furthermore, reinforcement learning algorithms consist of an intelligent agent that interacts with its environment and trains itself to make actions that maximize the cumulative reward [13].

The ascertaining study utilizes a supervised learning algorithm in order to predict which words and hence diseases and conditions that are most commonly associated with non-surviving patients, and to certain extent those surviving as well. In this case, words that are associated with deceased patients are labeled 1 and those associated with non-deceased ones with 0, making it a typical binary classification problem.

In order to implement this supervised binary classification, the data set needs to be subdivided into: (1) Training, (2) Validation and (3) Test data, as illustrated in Fig. 1. Initially, a fragment of the data set is extracted and marked as test data while the remainder is split into training and validation data [14]. The training data is used for fitting the parameters of the model while the test data is utilized for evaluating the final fit of the model trained from the training set. Lastly, the validation set is utilized for evaluating the final fit when optimizing the hyperparameters of the model [15].

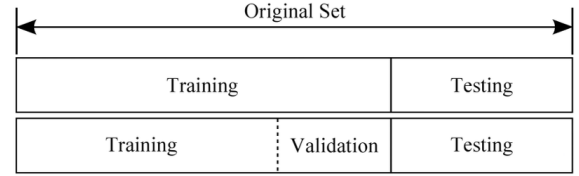


Fig. 1. Visual depiction of the subdivision of the original data set into training validation and test data. Illustration from Kacmajor, T [16].

Supervised machine learning algorithms are characterized by the following attributes: Internal parameters and hyperparameters as well as a cost function. Internal parameters are model-specific variables whose values can be evaluated from the training data. In regression models, these correspond to each respective coefficient and for a neural network, the biases b and weight matrices W . On the other hand, hyperparameters are variables whose values cannot be determined but serve the purpose of controlling the training process. Examples of such are for instance the learning rate for a neural network and the C hyperparameter for support vector machines. Finally, the cost function evaluates the performance of the model by calculating the error between predicted results and the original labeled variables. For optimal performance, the cost function needs to be minimized, which can be implemented by numerical iterative methods such as Stochastic Gradient Descent [17].

C. Natural language processing

1) *General*: Natural Language Processing (NLP) is field of study that lies within the intersection of artificial intelligence computer science and linguistics. It is concerned with programming algorithms that can process and analyze data generated from natural languages [18]. A natural language is a language that is in common use by humans, such as English or Swedish. More specifically it is of such kind that has evolved by natural means without any pre-planning or artificial constructions. On the other hand, processing more precisely refers to the methods of extraction of useful information from natural languages in order to create an algorithm that can derive meaning from it. Applications of NLP involve areas such as speech recognition, machine translation or in this particular case, *analysis of electronic health records*.

NLP is considered by many to be a difficult task. This mainly comes due to the fact that human language, spoken or written, is underspecified and ambiguous. Furthermore, correct interpretation of spoken or written language, requires knowledge about the surrounding and context of the particular situation [18]. This is extremely evident in the clinical situations, where health care providers must take into consideration that similar symptoms may manifest differently and arise from different conditions or diseases due to the unique set of circumstances and physiology of each individual patient. Such kind of ability is referred to as *differential diagnosis* and one can clearly conclude that an NLP mortality prediction algorithm does not take into account such aspects. For that reason, such algorithms cannot be fully correct and will misdiagnose certain numbers of patients. This is due to the fact that it cannot access the complete medical history of each respective individual from the data set [19].

2) *Computational models of languages*: Since machine learning algorithms specifically process numeric values and due to the fact that data generated from NLP comes in the form of words and sentences, two incompatible formats of data needs to be seamlessly integrated with each other. A few types of computational models can in such instance, rectify this issue and convert words into numerical format. More specifically this study will utilize the so called Bag-of-Words model (BOW). The premise of this approach is to construct a collection of all words in a document, listing their each respective frequencies [18]. As an example, the ICD-10 code for E13.351 correlates to the following diagnosis:

Other specified diabetes mellitus with proliferative diabetic retinopathy with macular edema

The first step is to conduct a so-called tokenization, whereby the sentence is split into tokens or in this case individual words [20]. Delimiters are in this case eliminated. As a result, the original sentence now becomes:

'Other', 'specified', 'diabetes',
'mellitus', 'with', 'proliferative',
'diabetic', 'retinopathy', 'with',
'macular', 'edema'

However, one may observe that since tokenization splits every single word in a sentence, it does not take into account words that rather should have been one token [21] such as in *diabetes mellitus*. Furthermore, grammatical inflections and suffixes are also not considered. In this instance, *diabetic* and *diabetes* should in actual case amount to the same token since the latter essentially corresponds to the genitive conjugation of the former. All of the aforementioned amounts to a limitation of the model when conducting the study and should of course be taken into consideration when evaluating the results. A BOW model can then be applied on this sentence and for instance, yield what has been shown in Tab. I. below:

TABLE I
WORD FREQUENCY MATRIX FOR E13.351

Count	Other 1	diabetes 1	neoplasm 0	coronary 0	with 2
	retinopathy 1	gastric 0	edema 1	renal 0	macular 1
	specified 1	proliferative 1	line 0	mellitus 1	...

3) *Logistic regression*: Utilizing logistic regression, a patient mortality classifier can be implemented upon the BOW-inputs of words from the diagnoses of each individual. The aim of this technique is to from a set of independent variables, predict the value of a dependent variable. The output must come in the form of a discrete value and a sigmoid function is to be fitted; which is the primary reason that logistic regression will be utilized. More specifically, a patient is either dead or alive, which entails that two maximum values (1 or 0) shall theoretically be assigned for each individual. In actuality, a *probabilistic* measure between the conditions (*alive or dead*) is assigned for each patient [22], [23].

Furthermore, the dependent variables are in this case each respective word from the BOW-inputs. Ideally, all data points should either be situated at the maximum and minimum of the sigmoid probability function: $\sigma : \mathbb{R} \rightarrow (0, 1)$, which is of the following form:

$$\sigma(X) = \frac{1}{1 + e^{-\Theta_k(X)}}, \quad (1)$$

where X is defined the set of independent variables of the N :th dimension and

$$\Theta(X) = \beta_0 + \beta_1 X_1 + \dots \beta_N X_N = \beta_0 + \sum_{n=1}^N \beta_n X_n, \quad (2)$$

a linear combination of each individual variable in X. Moreover, the logistic regression model aims to estimate the intercept β_0 on the Θ -axis, and parameters values $\beta_1 \dots \beta_N$ that optimize the fit. Since the relation between the dependent variable and parameters is non-linear a mean squared error loss cannot be utilized. Instead the optimal fit is obtained by minimizing the logistic loss (cost) function with log-likelihood. The logistic loss for a data point k can be interpreted as the probability of correlation between prediction p_k and outcome Θ_k . This is more specifically defined as $\log \sigma_k$ for $\Theta_k = 1$ and $\log(1 - \sigma_k)$ for $\Theta_k = 0$ [22], [24] and by amalgamation into the following expression:

$$\Theta_k \log \sigma_k + (1 - \Theta_k) \log(1 - \sigma_k), \quad (3)$$

the cross entropy, or the distribution of predicted and actual values is to be obtained [23]. Moreover, log-likelihood is equivalent to the sum of cross entropies for all data points k and defined as:

$$l = \sum_{k=1}^K (\Theta_k \log(\sigma_k) + (1 - \Theta_k) \log(1 - \sigma_k)), \quad (4)$$

with K defined as the total number of data points. As can be shown in eq. 5, the estimated parameters for optimal fit is obtained by maximizing l . This entails that the partial derivatives for the intercept β_0 and parameters $\beta_1 \dots \beta_N$ are to be set to zero [22], [24]:

$$0 = \nabla l = \begin{cases} \frac{\partial l}{\partial \beta_0} = \sum_{k=1}^K (\Theta_k - \sigma_k), \\ \ddots \\ \frac{\partial l}{\partial \beta_n} = \sum_{k=1}^K (\Theta_k - \sigma_k) X_n. \end{cases} \quad (5)$$

The optimal fit can also be estimated by maximizing the likelihood function (L), i.e. the probability of the independent variables being a function of the parameters:

$$L = \prod_{k:\Theta_k=1} \sigma_k \prod_{k:\Theta_k=0} (1 - \sigma_k), \quad \nabla L = 0. \quad (6)$$

This is referred to as a maximum likelihood estimation [22].

D. Model evaluation

1) *TF-IDF*: The aforementioned NLP algorithm predicts whether a patient will die based on EHR records but in order to yield interpretability, it would be of good use to also list the most common risk factors for patient mortality.

In such case, the Term Frequency-Inverse Document Frequency (TF-IDF) can be generated, which uses the frequency of set words in order to evaluate upon the importance of that specific word in a certain context. [25]. TF-IDF consists of two factors: term frequency, $tf(t, d)$ and inverse document frequency, $idf(t, D)$. Term frequency denotes the relative frequency of a specific term t in a document d and is more formally defined as function of t and d as shown below:

$$tf(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}. \quad (7)$$

As a clarification, $f_{t,d}$ denotes the raw count of a specific term while total number of terms in the document is represented by the $\sum_{t' \in d} f_{t',d}$ term. On the other hand, the inverse document frequency is an estimation of the informativeness of a specific word in all documents. This measurement is defined as:

$$idf(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}, \quad (8)$$

where N denotes the total document count of a corpus and $|\{d \in D : t \in d\}|$ the total number of document that a specific term t can be found [26]. This in turn means that the value for $idf(t, D)$ increases the rarer a term gets. TF-IDF is thus be expressed as:

$$tfidf(t, d, D) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \cdot \log \frac{N}{|\{d \in D : t \in d\}|}. \quad (9)$$

After calculating the TF-IDF value for each word in the data set, word importance for the positive and negative classes can be plotted and thus give an insight on what risk factors may most likely be a cause of patient mortality.

2) *Precision and recall*: A standardized method to assess the performance of a binary classification model is to subdivide the data set into four segments. Each segment is assigned with a label as shown below.

- **True Positives (TP)**: The positive class is correctly predicted by the model for one data element. The model has correctly predicted a patient that has died.
- **True Negatives (TN)**: The negative class is correctly predicted by a model for one element. The model has correctly predicted a patient that has survived.
- **False Positives (FP)**: The positive class is incorrectly predicted by a model for one element. The model has labeled a surviving patient as having died.
- **False Negatives (FN)**: The negative class is incorrectly predicted by a model for one element. The model has labeled a non-surviving patient as having survived.

In order to more intuitively understand labels, they can be visualized by a confusion matrix. For the aforementioned binary classification problem, the matrix is of a 2x2 dimension with each entries representing each respective label for the two classes [17]. As shown in Tab. II, these classes are referred to Class I and II for generalization. However a more suiting way of defining them for the study would be *surviving* and *non-surviving* or *positive* and *negative*. One should also note that the ordering of the labels in the matrix entries may vary depending on publication.

TABLE II
BINARY CONFUSION MATRIX

	Class 1	Class 2
Class 1	TN	FP
Class 2	FN	TP

The next step is to introduce performance metrics deriving from the previously stated accuracy metrics. For our binary classification algorithm, the following are commonly utilized:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}, \quad (10)$$

$$Recall = \frac{TP}{TP + FN}, \quad (11)$$

$$Precision = \frac{TP}{TP + FP}, \quad (12)$$

$$Specificity = \frac{TN}{TN + FP}, \quad (13)$$

$$F1_{Score} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}. \quad (14)$$

Accuracy refers to the proportion of correct predictions, *Recall (Sensitivity)* refers to the proportion of true positives in regard to the whole sample, *Precision* to the proportion of true positives in regard to all positives, and *Specificity* to the proportion of incorrect predictions. Finally, *F1-Score* refers to the harmonic mean of the *Recall* and *Precision* variables and gives a more holistic indication on the performance of the model [27].

3) *Receiver operating characteristic curve* : The True positive (TPR) and False positive rates (FPR) are defined as:

$$TPR = \frac{TP}{P}, \quad FPR = \frac{FP}{P}, \quad (15)$$

with the total population given by P .

Plotting TPR against FPR for each data point yields the ROC-curve (Receiver operating characteristic curve). For example, the (0,1) coordinate in the ROC-space is indicative of a perfect result and by connecting ever more data points, a graph resembling the ROC-curve is generated. Moreover, the area under the ROC-curve is defined as AUC and gives an estimation on the degree of separability of a model. Altogether, these serve as a visual estimation on the performance.

As can be observed in Appendix B [28], the highest (1, 1) and lowest obtainable values (0, 0) are connected by a diagonal line which is defined as the threshold for the classifier. Any coordinate further above from this diagonal is indicative of a more qualitative predictive result and vice versa. Furthermore the threshold line indicates that a classifier only does random guesses and is thus unable to make any discrimination.

AUC values fall within the range of $\{0, 1\}$ and greater values are indicators of better performance. However as AUC approaches 0.5, ROC-curve converges to the threshold line and the model seemingly makes no discrimination between the two classes [29]. In other words it cannot distinguish between surviving and non-surviving patient and thus, 0.5 will be used as a threshold for evaluation.

III. METHODS

A. Dataset

1) *Background*: The MIMIC-IV Clinical Database consists of clinical information from 523,740 patients, extracted from either hospital EHR as well as records from the ICU units at the Beth Israel Deaconess Medical Center (BIDMC). In order to maintain anonymity, each patients were assigned a unique numeric identifier, *subject_id* with all dates being randomly shifted into the future. As a result, two patients may not have been admitted in the same years even if the database suggests otherwise. However, event are consistent [3].

The data within MIMIC-IV is stored in CSV tables and subdivided into three modules: *core*, *hosp* and *icu*. Each table contains several columns with patient-specific identifiers. The *core* module provides patient identification information and most importantly, contains three tables: *patients* which contains patient demographics records, *admissions* which stores hospitalization records such as admission and discharge dates, and *transfers* that provides information about each ward stay for each admission. The *hosp* module provides EHR data and most importantly contains table with hospital billing information, medication administration and laboratory measurements among others. Lastly, the *icu* module contains information extracted from the intensive care units at the BIDMC. As an additional note, one should also take into account that data derived from MIMIC-IV may contain idiosyncrasies since such are commonplace in conventional clinical practice [3].

2) *Implementation*: In this study, the following MIMIC tables in the *core* and *hosp* modules were utilized:

- **admissions (core)**: Apart from *subject_id* identifying each respective patient, other relevant columns in this table are the following:
 - **hadm_id**: Numerical identifier for each hospital admission.
 - **admittime**: Date of admission for patient in the format of: YYYY-MM-DD hh:mm:ss
 - **admittype**: Type of admission for each patient. *Elective*, *emergency*, *newborn* or *urgent*.
 - **dischtime**: Discharge date for patient in the aforementioned format. If patient has survived, this column will be empty.
 - **deathtime**: Death date for patient in the aforementioned format.
- **diagnoses_icd (hosp)**: Table listing each diagnosis for each subject (*patient*) as a numerical code in ICD-9 format. Patients are identified by their specific *subject_id* and *hadm_id*.
- **d_icd_diagnoses (hosp)**: Conversion table matching each ICD-9 code to the long title for a specific diagnosis. For example, 07953 in pure code correlates to "Human immunodeficiency virus, type 2 [HIV-2]"."

Due to the restrictive nature of the dataset and ethical concerns, patient data that has been shown are modified and artificial and do not conform to the original raw data. Instances of *subject_id* and *hadm_id* that have been listed are completely fictional and cannot be found in any of the MIMIC-IV tables.

B. Data pipeline overview

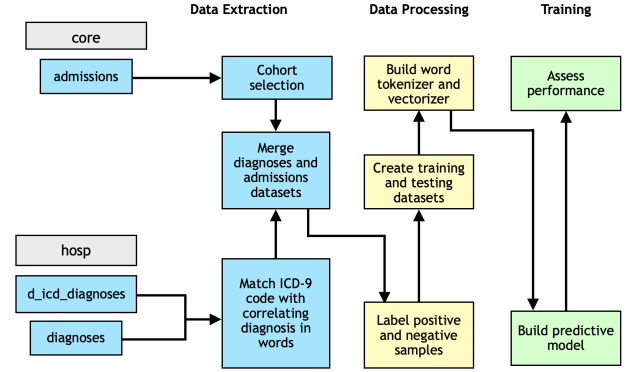


Fig. 2. Visualization of the data pipeline for the study. As a clarification, procedures for each phase have been assigned a specific color. It has also clearly seen which tables fall under which modules as shown in the *Data Extraction* segment.

1) *Cohort selection*: Initially, the *admissions* table was loaded using the *pandas* dataframe, and the dates under the *admittime* column converted from string to datetime format. Subsequently, the specific cohort, i.e. patients that had died in-hospital had to be identified. This was done calculating the time between hospital admission and death for deceased patients, which automatically creates a selection of the specific cohort. Out of 523,740 patients, 9,337 had died in-hospital and the distribution for the duration between admission and death for this cohort can be seen in Fig. 3:

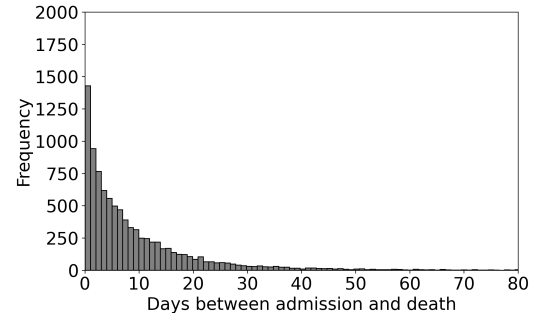


Fig. 3. Distribution for duration between admission and death in days.

2) *ICD-9 to text conversion*: Entries in the *diagnoses_icd* table may look like the Tab. III:

TABLE III

subject_id	hadm_id	seq_num	icd_code	icd_version
12345678	87654321	3	0010	9
46372930	10384756	3	M99	9
00392482	56704039	3	V3600	9

In this case, each entry under the *icd_code* column was matched and replaced with its corresponding text title under the *long_title* column in the *d_icd_diagnoses* table. As for clarification, Tab. IV provides an except of the first 7 rows from the aforementioned table.

TABLE IV
EXCERPT FROM THE D_ICD_DIAGNOSES TABLE

icd_code	icd_version	long_title
0011	9	Cholera due to vibrio cholerae el tor
0019	9	"Cholera, unspecified"
0020	9	Typhoid fever
0021	9	Paratyphoid fever A
0022	9	Paratyphoid fever B
0023	9	Paratyphoid fever C

3) *Merging of datasets*: Since the cohort selection was made in the *admissions* table, the *icd_code* column in the *diagnoses_icd* table had to be merged with it and each individual text diagnosis inserted. As a precautionary measure, the possibility for duplications and missing diagnoses for admissions were inquired. It was discovered that 0.5 % of diagnoses were missing and this group of subjects would not have been processed by the algorithm.

C. Data Processing

The data was split in to training, testing and validation set as specified in Fig 1. Furthermore, subjects were labeled either as positive and negative, with the former referring to non-surviving and the latter with surviving patients. As such the following output labels: ($1 = died$, $0 = survived$) were assigned for each individual in the dataset. A quick parsing in the training set showed the following:

TABLE V
INSTANCES OF POSITIVE AND NEGATIVE SAMPLES

Description	Instances
Number of positive samples	8,974
Number of negative samples	514,946
Number of samples	523,740
Total number of lines	5,280,351

This gave an indication that the dataset was heavily imbalanced. In order to balance in the training set, the negative samples were sub-sampled. This constituted the most fitting course of action according to Fithan et al. [30].

D. Training

The NLP model utilized for the analysis and processing of the data was borrowed from Ameisen, E. [31] and comprised a BOW algorithm. Using the *word_tokenizer* function from the *nltk* package, each sentence of the diagnoses was tokenized. The tokenized sentences were subsequently fitted to *CountVectorizer* from *sklearn* in order to learn as well as calculating the frequency of each word. With this measure, the most common words for non-survived patients were to be extracted and plotted, as can be shown in Appendix A. The number of words included is regulated by the *max_features* parameter (*hyperparameter*) of the *CountVectorizer* tool. Naturally, conjugations, prepositions and conjunctions etc. are over-represented and have to be filtered out. These types of words are referred to as stop words [32] and a list of such derived from the word frequency analysis and the *Oxford English Corpus* was constructed.

The text from the patient diagnoses was then converted into a numerical matrix format. The mortality prediction model was built upon a logistic regression model. More specifically, the *LogisticRegression* tool from *sklearn* was utilized and by adjusting its *C* hyperparameter the model could either give more or less weight to the training set (*which will be further elaborated upon in the Results section*). This constituted the final step in constructing the mortality prediction algorithm.

Algorithm 1 Pseudocode for BOW implementation [33]

```

1: Initialize null vector, WordCounts = [0,0,...0]
2: for token in tokenized text do
3:   if token in dict then
4:     Get dict index of token
5:     WordCounts[token_index] ++
6:   else
7:     continue
8:   end if
9: end for
10: return WordCounts

```

IV. RESULTS

A value of $C = 0.1$ was chosen for the regularization hyperparameter and the world limit for the vectorizer (*max_features*) was set at 1000. Lastly, the threshold had been set at AUC = 0.5 and the the performance is visualized in Fig. 4:

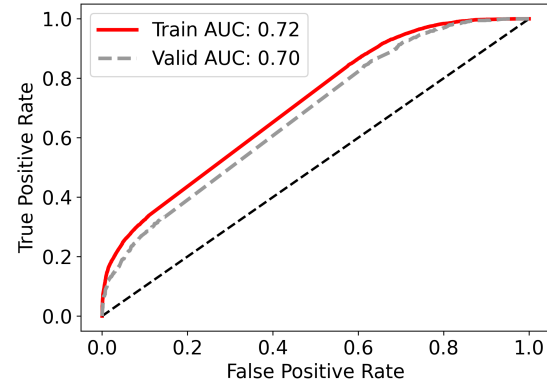


Fig. 4. ROC curve for $C = 0.1$. The dotted black line is labeled as the threshold set at AUC = 0.5. This entails that AUC = 1 gives indication to a perfect classifier while AUC = 0 signifies that no sample have been labeled correctly. At AUC = 0.5, the predictor makes random guesses.

TABLE VI
PERFORMANCE METRIC FOR MORTALITY PREDICTION ALGORITHM

	Training	Validation
AUC	0.722	0.697
Accuracy	0.651	0.584
Recall	0.697	0.673
Precision	0.638	0.027
Specificity	0.604	0.582
F1-Score	0.697	0.052

The performance of the mortality classification model can be observed from Tab. VI. A training AUC of 0.722 and validation AUC of 0.697 were yielded.

Furthermore, a confusion matrix was generated in order to more intelligibly visualize the variance between the performance metric rates.

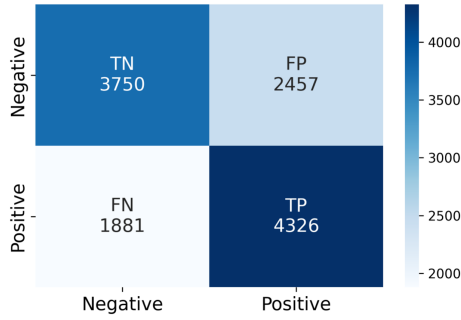


Fig. 5. Confusion matrix for mortality prediction algorithm

As seen in Fig. 5, a graphical representation was utilized to visualize the numerical frequency of each accuracy metric. Higher frequencies are thus represented by the darker shades of colors and lower frequencies with brighter ones.

Furthermore, the dependence on hyperparameters for the model had been visualized in Figs. 6 and 7 by plotting each variable against the AUC values.

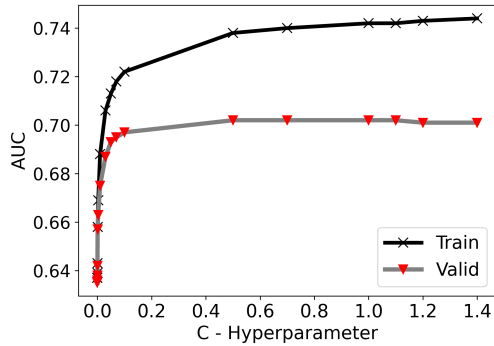


Fig. 6. Dependence of the regularization constant, C for model performance (AUC). As previously stated, C was set at 0.1.

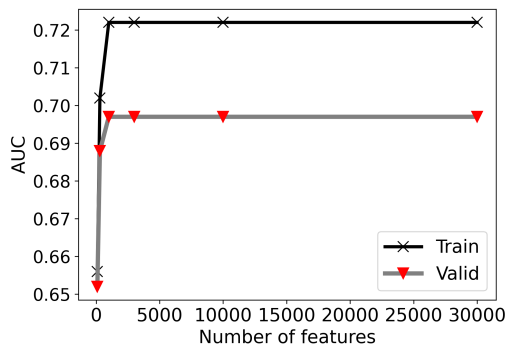


Fig. 7. Dependence of word limit for vectorizer ($max_features$) for model performance (AUC). As previously stated, $max_features$ was set at 1000.

Finally, the risk factors for patient mortality as well as survival were plotted in order of importance by calculating the TF-IDF scores for each word in corpus. Tab. VII lists the 50 most important factors or more specifically, words for the positive (non-surviving) and negative (surviving) subsets of the population.

TABLE VII
RISK FACTORS FOR PATIENT MORTALITY IN ORDER OF IMPORTANCE

Negative		Positive	
Word	Score	Word	Score
abuse	0.88	septicemia	1.139
pain	0.811	shock	1.069
delivered	0.801	severe	0.865
prophylactic	0.745	arrest	0.746
chest	0.722	acidosis	0.71
inoculation	0.716	cardiac	0.685
disorder	0.712	encephalopathy	0.683
suicidal	0.71	hemorrhage	0.68
ideation	0.71	sepsis	0.646
vaccination	0.654	cerebral	0.575
nontraffic	0.625	necrosis	0.575
condition	0.564	nonmotor	0.54
antepartum	0.564	failure	0.535
anxiety	0.54	pulmonary	0.532
depressive	0.449	vehicle	0.516
obesity	0.448	neoplasm	0.511
syncope	0.438	traffic	0.511
alcohol	0.426	respiratory	0.51
diarrhea	0.415	ventricular	0.483
asthma	0.406	vascular	0.48
colon	0.395	liver	0.471
hepatitis	0.393	ascites	0.464
abscess	0.392	coagulation	0.455
motorcycle	0.392	hyperpotassemia	0.454
motor	0.389	hypoxemia	0.453
single	0.382	hemiplegia	0.433
mention	0.379	edema	0.411
postoperative	0.376	iv	0.403
reflux	0.375	hyperosmolality	0.399
apnea	0.375	hypernatremia	0.399
joint	0.372	hydrocephalus	0.379
primary	0.371	bronchus	0.376
viral	0.364	vomit	0.376
myelopathy	0.363	pneumonitis	0.376
hyperlipidemia	0.356	inhalation	0.376
dehydration	0.35	food	0.376
leg	0.347	car	0.374
psychosis	0.346	acute	0.365
nausea	0.345	convulsions	0.359
calculus	0.336	pneumonia	0.358
schizophrenia	0.323	septic	0.357
carrier	0.321	fibrillation	0.344
lumbago	0.32	cardiogenic	0.339
appendicitis	0.319	anticoagulants	0.337
driver	0.313	intracerebral	0.323
cesarean	0.307	pressure	0.316
vomiting	0.305	thrombocytopenia	0.311
thyroid	0.303	stated	0.294
period	0.302	brain	0.292
perinatal	0.302	peripheral	0.287

As can be noted, patients that had been hospitalized due to septic- and infectious related causes, acute conditions, as well as deficiencies in the central nervous, circulo-respiratory and hematological systems had the highest mortality rates upon admission. On the contrary, individuals admitted for child delivery, pain management, immunization as well as psychological conditions were most likely to survive.

The classification threshold had been set at $AUC = 0.5$ but performance metrics for additional values were also analyzed and plotted, as can be seen in Fig. 8. As a clarification, classification thresholds do not impact the AUC, which in turn has not been plotted.

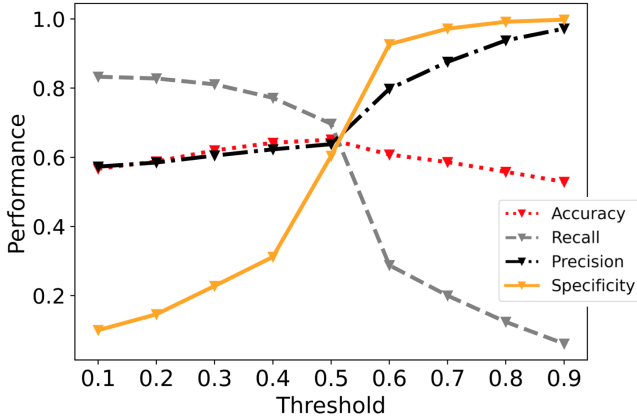


Fig. 8. Dependence on classification threshold values for performance metrics in training data.

V. DISCUSSION

A. Model performance

The mortality prediction algorithm demonstrated a training AUC of 0.72 and validation AUC of 0.70. As stated by Tab. VIII, it would have *good* discrimination. However there is a drastic discrepancy between the precision metrics of the two sets, with the validation set only exhibiting a precision score of 0.027. This is attributed to the fact that the training set has been balanced while the samples in the validation set still adheres to the original distribution.

TABLE VIII
MODEL PERFORMANCES FOR AUC-VALUES [29]

AUC (0-1)	Degree of agreement
0.9 - 1.0	Outstanding
0.8 - 0.9	Excellent
0.7 - 0.8	Good
0.6 - 0.7	Satisfactory
0.5 - 0.6	Poor
≤ 0.5	No discrimination

Tab. VI also shows that the validation set yields worse performance in regards to all other metrics as compared to the testing set. A possible explanation could be the fact that ICD-diagnoses are highly condensed; not able convey the full situation and concurrent conditions for each individual patient. While training, the algorithm is able to gain a deep insight into mortality risk single text (*corpus*).

As a result a substantial amount of insight can be gained with proportionally less computing power. When validating, the algorithm can only derive its judgment from one sentence of information, thus not being able to take into account the complex interplay of factors as in the training case. This implies that mortality rates for patients with more lethal conditions can be sufficiently predicted with less amount of information. At the same time, patient mortality for less life-threatening conditions is more difficult to predict. As an example, septic-related causes were identified as the most significant risk factors for patient mortality, and the severity for such conditions has repeatedly been confirmed by previous studies [34]. Consequently, the predicted mortality rate for this group of patient will be high regardless of other conditions that they may have. The aforementioned phenomenon may also give an explanation into why the model demonstrates a preference for predicting non-surviving as opposed to surviving patient correctly. As can be observed in Fig. 5, there is a higher frequency for true positives than for true negatives as shown in Tab. VI, it can clearly be deduced that the recall rates are greater than all other performance metrics. As a result, positive risk factor are a more reliable indication for mortality than negative risk factors being for survival.

It must finally be reiterated that the mortality prediction metric is given as a decimal within the range of 0 to 1 and thus, only indicates a *probability*; not an exact statement. As a result there is a certain degree ambiguity in a substantial amount of the predictions, which needs to be considered. Some patients labeled as *died* may have been assigned with a value within the range of 0.5 and could as likely have survived since the the assigned decimal value is rounded to its nearest integer.

B. Dependence on hyperparameters

As can be observed from Figs. 6 and 7, the model is not particularly susceptible for any alterations of starting at $C \approx 0.1$ and *max_features* from 500-1000. At these values, a convergence behavior can be seen to have emerged, resulting in some degree of overfitting. Under such circumstances, final performance of the model is compensated due to the fact that it has overlearned the noise and details of the training data [35]. The values of the hyperparameters have thus been chosen such that AUC is maximized while simultaneously not falling within the region where the curves plateau. In summary it can be concluded that the model is more susceptible for variations of the C hyperparameter than word limit for the vectorizer.

C. Dependence on classification threshold

As can be noted from Fig. 8, increasing classification thresholds result in increased precision and specificity, as well as decreased recall. Simultaneously, accuracy increases for thresholds lesser than 0.5, while decreasing for values from 0.5 to 1. However, with incremental margins. This can for the most part be attributed to the fact both true positive and false negative rates are reduced when the threshold is raised, which can be verified by plugging into eqs. 10 - 14.

D. Mortality risk factors

1) *Insights*: By analyzing the mortality risk factors in Tab. VII, it can be concluded that disturbances of internal physiological systems result in considerably higher fatalities and are thus over-represented in non-surviving patient. Most importantly, it can be concluded such patients are experiencing that acute conditions or undergoing an unexpected event such as *shock* or *failure*. These groups of patients should be prioritized for admission to intensive care units. Moreover, surviving patient were more likely admitted either due to a specific injury, a daily inconvenience or for a specific task such as *vaccination*, *child delivery* and *psychiatric treatment*. It can be argued that this group of patient should rather have been placed into general hospital wards and not utilize resources that would have been allocated to more urgent cases.

2) *Constraints*: By plotting the most important risk factors for patient mortality a deep insight can be gained. Most importantly, it has the potential serving as a decision support system, giving suggestions on how to most efficiently allocate patients and resources in clinical situations. However it must be noted that this comes at the expense of numerous limitations. The data was collected from one specific hospital in the United States before the advent of the COVID-19 pandemic. It is therefore uncertain whether the results from the study can be used as a reference point for other hospitals within or outside of the United States, such as in Sweden.

Although basic human anatomy is universal, cultural and environmental factors may significantly impact the mortality assessment by the algorithm. It is thus difficult to assess whether the (TF-IDF) score for a given risk factor is the result of a behavioral pattern of the population or natural causes that can be explained with medical research.

Sepsis and related causes were placed with great importance due to physiological reasons but factors such as *Pain* are extremely ambiguous and can hugely vary in severity. Likewise, It is also impossible to assess the impact of the SARS-CoV-2 virus by only utilizing data from the MIMIC-IV database, unless an updated version with data from COVID-19 patients were to be released. This is due to the fact that SARS-CoV-2 virus has clearly disrupted the behavioral patterns of populations and may interfere with previous health patterns for each individual.

E. Technical limitations

Due to technical limitations not all patient diagnoses were from ICD-9 format to text, which means that more nuances could have been gained, conceiving a more accurate model. For these individuals, the original ICD-9 code remained unaltered when processed and their potential contributions were thus neglected. Most importantly, this would relieve the uncertainty of predicting outcomes for patients diagnosed with more ambiguous conditions.

In order to yield a more representative assessment of the model, physician notes could have been used as the validation set. Physician notes are however only included in MIMIC-III [7] and due to their lengths as compared to the ICD-9 codes, a considerable amount of processing power would have been required.

VI. CONCLUSIONS

It can be concluded that the MIMIC-IV database gives a stable foundation for a hospital clinical decision support system; able to give a considerable amount of insight for health care workers. Some of the identified risk factor may seem self-evident. However, other factors may significantly alter previous comprehensions on the interplay of etiological factors. Most importantly, it is hoped that the results from this study provide suggestions on which types patients to prioritize in order to most efficiently allocate resources. However, the results are not fully accurate and reliable and precaution must be taken when interpreting.

This study has also shown that it is possible to yield interpretability from EHR databases such as MIMIC-IV and that relevant information can be extracted from seemingly arbitrary variables. However, this comes in the pretext of being specific and having a clear outline and which actions to take and how to interpret the data. Terms such as patient mortality and mortality risk factors must be clearly defined and if being interchanged, it can significantly impact the clinical assessment in a certain situation. Furthermore, it can be seen that the MIMIC databases have a high degree of versatility and reproducibility. Many modifications and variations can be made upon a pre-existing pipeline and there is much room for further research to be made upon what has been established in this study.

Hyperparameters, technical limitations and aspects of the data pipeline may significantly affect the accuracy of the predictions. The hyperparameters must be adjusted precisely in order to both maximize efficiency as well as performance, and it has been shown that utilizing ICD codes for mortality prediction may be more relevant when training the classifier as opposed to validation. More specifically, it could be noted that the classifier had higher rates of predicting true positives than true negatives correctly.

VII. FUTURE WORK

It would be of interest to incorporate physician notes in to the study. This can either serve as a more representative way of validating the model both but also as a way of comparison between the ICD-based approach used in this study with the procedures of previous articles which have utilized physician notes as a foundation for their studies [36]. Moreover, it would be of interest to conduct the same study with MIMIC-III and compare the performances from the newer and older versions of the database.

The current model can also be expended upon if other and more advanced NLP and machine learning techniques could be utilized and perhaps be incorporated and bring potential improvements. Likewise, more performance metrics could have been analyzed and taken into consideration in order to yield a deeper and more holistic assessment on the performance. Finally it would be of great interest to conduct the study without the aforementioned technical limitations and evaluate what impact this factor may pose.

ETHICS STATEMENT

All patient information in the dataset is anonymized and any indications that may be linked to specific individuals have been eliminated or omitted. Furthermore, the dataset which can be accessed on *PhysioNet* is restricted, and only credentialed users that have signed a data use agreement and completed the *Data or Specimens Only Research* course by the *Collaborative Institutional Training Initiative (CITI)* program are granted access. Upon credentialing, an approved motivation for the usage of the resources is also necessitated.

APPENDIX A

MOST FREQUENT WORDS IN TRAINING DATA

APPENDIX B

EXPLANATION OF THE ROC-CURVE

ACKNOWLEDGEMENTS

First and foremost, the author would like to express gratitude to Ragnar Thobaben for his continuous support, guidance and advice during the course of the project. His insightful comments have been tremendously meaningful and encouraging. The author is also appreciative towards the team at the *Laboratory for Computational Physiology* at Massachusetts Institute of Technology (MIT) for the dedication that they have shown in realizing the MIMIC-database and accompanying resources. Their efforts have significantly expanded upon the opportunities within the intersection between healthcare and technology, and it is of great interest to observe the further innovations that may arise. Lastly, the author would like to thank his family, friends and classmates. Without their encouragement, support and trustful attitudes, it would have been impossible to conduct this study.

REFERENCES

- [1] P. Kubben, M. Dumontier, and A. Dekker, *Fundamentals of Clinical Data Science*. Cham Switzerland: Springer Open, 2019.
- [2] S. Wang, M. B. McDermott, G. Chauhan, M. Ghassemi, M. C. Hughes, and T. Naumann, "Mimic-extract," *Proceedings of the ACM Conference on Health, Inference, and Learning*, p. 1, 2020.
- [3] A. Johnson, L. Bulgarelli, T. Pollard, S. Horng, L. A. Celi, and R. Mark. (2021, Mar) Mimic-iv. [Online]. Available: <https://physionet.org/content/mimiciv/1.0/>
- [4] T. Gentimis, A. J. Alnaser, A. Durante, K. Cook, and R. Steele, "Predicting hospital length of stay using neural networks on mimic iii data," *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress*, pp. 1194–1201, 2017.
- [5] A. E. W. Johnson, T. J. Pollard, and R. G. Mark, "Reproducibility in critical care: a mortality prediction case study," in *Proceedings of the 2nd Machine Learning for Healthcare Conference*, ser. Proceedings of Machine Learning Research, F. Doshi-Velez, J. Fackler, D. Kale, R. Ranganath, B. Wallace, and J. Wiens, Eds., vol. 68. PMLR, 18–19 Aug 2017, pp. 361–376.
- [6] J. Parreco, A. Hidalgo, R. Kozol, N. Namias, and R. Rattan, "Predicting mortality in the surgical intensive care unit using artificial intelligence and natural language processing of physician documentation," *The American Surgeon*, vol. 84, no. 7, p. 1190–1194, 2018.
- [7] A. Johnson, T. Pollard, and R. Mark. (2016, Mar) Mimic-iii. [Online]. Available: <https://physionet.org/content/mimiciii/1.4/>
- [8] S. J. Russel and P. Norvig, *Artificial Intelligence A Modern Approach*. Englewood Cliff, NJ: Prentice Hall, 1995.
- [9] J. R. Koza, F. H. Bennett, D. Andre, and M. A. Keane, "Automated design of both the topology and sizing of analog electrical circuits using genetic programming," *Artificial Intelligence in Design '96*, p. 151–170, 1996.
- [10] T. Guo, J. Dong, H. Li, and Y. Gao, "Simple convolutional neural network on image classification," in *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, 2017, pp. 721–724.
- [11] J. Miao and W. Zhu, "Precision-recall curve (prc) classification trees," *Evolutionary Intelligence*, 2021.
- [12] M. Borovcnik, H.-J. Bentz, and R. Kapadia, "A probabilistic perspective," *Chance Encounters: Probability in Education*, vol. 12, p. 27–71, 1991.
- [13] D. Johnson. (2022, Mar) Reinforcement learning: What is, algorithms, types amp; examples. [Online]. Available: <https://www.guru99.com/reinforcement-learning-tutorial.html>
- [14] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning with applications in R*. New York, NY: Springer, 2013.
- [15] J. Brownlee. (2020, Aug) What is the difference between test and validation datasets? [Online]. Available: <https://machinelearningmastery.com/difference-test-validation-datasets/>
- [16] T. Kacmajor. (2016, may) Svm model selection – how to adjust all these knobs pt. 2. [Online]. Available: <https://tomaszkacmajor.pl/index.php/2016/05/01/svm-model-selection2/validation-set/>
- [17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. Cambridge, MA: MIT Press, 2016.
- [18] J. Boye, "Natural language processing," in *DD2380 Artificial Intelligence*, KTH, Stockholm, Lecture notes, Sep 2021.
- [19] D. Lamba, W. H. Hsu, and M. Alsadhan, "Predictive analytics and machine learning for medical informatics: A survey of tasks and techniques," *Machine Learning, Big Data, and IoT for Medical Informatics*, p. 1–35, 2021.
- [20] S. Vajjala, B. Majumder, A. Gupta, and H. Surana, *Practical natural language processing: A comprehensive guide to building real-world NLP systems*. Newton, MA: O'Reilly, 2020.
- [21] D. Lopez Yse. (2019, Apr) Your guide to natural language processing (nlp). [Online]. Available: <https://towardsdatascience.com/your-guide-to-natural-language-processing-nlp-48ea2511f6e1>
- [22] D. A. Freedman, *Statistical models: Theory and practice*. New York: Cambridge University Press, 2012.
- [23] M. Nielsen, *Neural Networks and Deep Learning*. Determination Press, 2015. [Online]. Available: <https://books.google.se/books?id=STDBswEACAAJ>
- [24] D. G. Kleinbaum and M. Klein, *Logistic regression: A self-learning text*. New York, NY: Springer, 2011.
- [25] J. Beel, B. Gipp, S. Langer, and C. Breiteringer, "Research-paper recommender systems: A literature survey," *International Journal on Digital Libraries*, vol. 17, no. 4, p. 305–338, 2015.
- [26] R. P. Manning, C.D and H. Schutze, "Scoring, term weighting, and the vector space model," *Introduction to Information Retrieval*, p. 100, 2008.
- [27] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing Management*, vol. 45, no. 4, pp. 427–437, 2009.
- [28] D. Cowan. (2020) Confusion matrix. [Online]. Available: <https://www.ml-science.com/confusion-matrix>
- [29] S. Yang and G. Berdine, "The receiver operating characteristic (roc) curve," *The Southwest Respiratory and Critical Care Chronicles*, vol. 5, no. 19, p. 34, 2017.
- [30] W. Fithian and T. Hastie, "Local case-control sampling: Efficient subsampling in imbalanced data sets," *The Annals of Statistics*, vol. 42, no. 5, oct 2014.
- [31] E. Ameisen. (2021, Jan) Concrete solutions to real problems. [Online]. Available: https://github.com/hundredblocks/concrete_NLP_tutorial/blob/master/NLP_notebook.ipynb
- [32] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Statistical models: Theory and practice*. Cambridge: Cambridge University Press, 2022.
- [33] M. Maktabar, A. Zainal, M. A. Maarof, and M. N. Kassim, "Content based fraudulent website detection using supervised machine learning techniques," *Hybrid Intelligent Systems*, p. 294–304, 2018.
- [34] R. S. Hotchkiss, L. L. Moldawer, S. M. Opal, K. Reinhart, I. R. Turnbull, and J.-L. Vincent, "Sepsis and septic shock," *Nature Reviews Disease Primers*, vol. 2, no. 1, pp. 1–5, 2016.
- [35] R. Roelofs, V. Shankar, B. Recht, S. Fridovich-Keil, M. Hardt, J. Miller, and L. Schmidt, "A meta-analysis of overfitting in machine learning," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019, pp. 1–11.
- [36] A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Marcus, M. Sun, and et al., "Scalable and accurate deep learning with electronic health records," *npj Digital Medicine*, vol. 1, no. 1, pp. 1–18, 2018.

Investigation of Information-Theoretic Bounds on Generalization Error

Kevin Pettersson and Reza Qorbani

Abstract—Generalization error describes how well a supervised machine learning algorithm predicts the labels of input data that it has not been trained with. This project aims to explore two different methods for bounding generalization error, f -CMI and ISMI, which explicitly use mutual information. Our experiments are based on the experiments in the papers in which the methods were proposed. The experiments implement and validate the accuracy of the mathematically derived bounds. Each methodology also has a different method for calculating mutual information. The ISMI bound experiment used a multivariate normal distribution dataset, whereas a dataset consisting of cats and dogs was used for the experiment using f -CMI. Our results show that both methods are capable of bounding the generalization error of a binary classification algorithm and provide bounds that closely follow the true generalization error. The results of the experiments agree with the original experiments, indicating that the proposed methods also work for similar applications with different datasets.

Sammanfattning—Generaliseringsfel beskriver hur väl en övervakad maskininlärnings algoritm förutspår etiketter av indata som den inte har blivit tränad med. Syftet med projektet är att utforska två olika metoder för att begränsa generaliseringsfelet, f -CMI och ISMI som explicit använder ömsesidig information. Vårt experiment är baserat på experimenten i artiklarna som tog fram metoderna. Experimenten implementerade och validerade noggrannheten av de matematiskt härledda gränserna. Varje metod har olika sätt att beräkna den ömsesidiga informationen. ISMI gräns experimentet använde en flerdimensionell normalfördelning som data set, medan en datauppsättning med katter och hundar användes för f -CMI gränsen. Våra resultat visar att båda metoder kan begränsa generaliseringsfelet av en binär klassificerings algoritm och förse gränser som nära följer det sanna generaliseringsfelet. Resultatet av experimenten instämmer med de ursprungliga författarnas experiment vilket indikerar att de föreslagna metoderna också fungerar för liknande tillämpningar med andra data set.

Index Terms—Generalization error, ISMI, functional conditional mutual information, Generalization bound

Supervisors: Amaury Gouverneur

TRITA number: TRITA-EECS-EX-2022:171

I. INTRODUCTION

The field of machine learning (ML) has been in the spotlight in recent years, and the vast amount of research conducted in the field has led it to grow exponentially compared to other related fields. The growing interest in machine learning could be in part due to the fact that it is a new technology that has changed our perception of what computers are capable of doing, but the main driving factor is that it has been proven to be a more capable solution for a subset of important problems, for example, determining the 3D structure of proteins [1]. As

supervised learning algorithms are used more and more in critical areas, such as in social welfare systems [2], it becomes more important to measure how well a supervised machine learning model will perform when it is working with data that it has not been trained on, i.e. how well the algorithm generalizes. More specifically, generalization error is a metric that measures how well an algorithm predicts the output values when the input is unseen data, i.e. data that was not used when training the algorithm. While the definition of generalization error is quite straightforward, calculating the generalization error exactly is a difficult task because the distribution of the underlying population (of data) is often unknown, and thus the goal is to find an estimate for the upper bound of the generalization error instead.

II. PROBLEM DESCRIPTION

The task of estimating an upper bound for the generalization error, hereafter called the generalization bound, has gained lots of attention, and several different methods have been proposed for estimating the generalization bound [3], [4]. This project aims to explore, and compare two proposed methods, f -CMI [3] and ISMI [4], that explicitly uses mutual information for estimation of the generalization bound. Using mutual information for estimating generalization bounds is a new approach in this research field and it has shown promising results compared to earlier approaches. Different generalization bound estimation methods explored in this project are in the form of mathematical inequalities that both use mutual information calculated between two random variables in some capacity. There are different methods for estimating mutual information itself, as variables can take on different kinds of values depending on which aspect of the model and data we are using. We use the concept of functional conditional mutual information as described in [3] to calculate the mutual information for the f -CMI bound. The ISMI bound instead uses an estimator called the bias-improved-KSG estimator for calculating the mutual information, that was proposed in [5].

III. PRELIMINARIES

This section aims to give a theoretical overview of fundamental concepts related to this project. This section starts with a theoretical overview of mutual information and generalization error then the algorithms used to determine the mutual information and generalization bounds are described. Note that all logarithms are natural logarithms unless stated otherwise.

A. Mutual information

In order to describe what mutual information is we have to first introduce the concept of entropy as described in [6]. Let X be

a discrete random variable with the probability mass function $p_X(x) = Pr\{X = x\}$ where $x \in \chi$. Then the entropy of a discrete variable X can be written as

$$H(X) = - \sum_{x \in \chi} p(x) \log p(x) \quad (1)$$

Let X, Y be a pair of discrete random variables with a joint distribution $p(x, y)$. The joint entropy $H(X, Y)$ is defined as

$$H(X, Y) = - \sum_{x \in \chi} \sum_{y \in \Upsilon} p(x, y) \log p(x, y) \quad (2)$$

The relative entropy, also called Kullback-Leibler distance, between two probability mass functions $p(x)$ and $q(x)$ is a quantity which measures the difference between two probability distributions. The relative entropy is defined as

$$D(p||q) = \sum_{x \in \chi} p(x) \log \frac{p(x)}{q(x)} \quad (3)$$

If X and Y are two random variables then the mutual information between X and Y is a measure of the amount of information that one random variable contains about another random variable. It can also be interpreted as how much uncertainty of a random variable decreases when there is knowledge of the other random variable. Let $p(x)$ and $p(y)$ be the marginal probability mass functions between X and Y and $p(x, y)$ the joint distribution between the two variables. Mutual information $I(X; Y)$ between two random variables X and Y is defined as

$$I(X; Y) = \sum_{x \in \chi} \sum_{y \in \Upsilon} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \quad (4)$$

Comparing equation 4 with equation 3, we can describe mutual information as the relative entropy between the joint distribution $p(x, y)$ and the product distribution $p(x)p(y)$. A more detailed overview of the above theory can be found in [6].

B. Generalization error

Let μ be an unknown distribution over a known set \mathcal{Z} with n elements. Consider an input sequence $Z = \{Z_1, Z_2, Z_3, \dots, Z_n\} \sim \mathcal{Z}^n$ of n independent and identically distributed random variables $Z_i \in \mathcal{Z}$. A supervised learning algorithm takes as input the set Z and produces a hypothesis $W \in \mathcal{W}$ according to the conditional distribution $P_{W|Z}$. The supervised learning algorithm is essentially a random mapping from the set Z to W . A metric used for how well the algorithm predicts a given sample Z_i is the loss function $l : \mathcal{W} \times \mathcal{Z} \rightarrow \mathbb{R}^+$. A learning algorithm's main task is to choose a $w \in \mathcal{W}$ that minimizes the population risk

$$L_\mu(w) \triangleq \mathbb{E}_{Z \sim \mu} [l(w, Z)] \quad (5)$$

But since the distribution μ is usually unknown, the population risk cannot be directly calculated. Instead we measure the performance of the algorithm by calculating the empirical risk

$$L_Z(w) \triangleq \frac{1}{N} \sum_{i=1}^N [l(w, Z_i)] \quad (6)$$

on the training dataset Z . Using this definition we can define the generalization error as the Expected difference between the population risk and the empirical risk of the output hypothesis

$$gen(\mu, P_{W|Z}) \triangleq \mathbb{E}_{W, Z} [L_\mu(W) - L_Z(W)] \quad (7)$$

The expectation is taken over the joint distribution $P_{W, Z} = P_Z \otimes P_{W|Z}$.

C. BI-KSG estimator

The BI-KSG (bias improved) estimator is a modified version of the original KSG estimator proposed in [7]. The KSG estimator is one of the most popular estimators for estimating mutual information from i.i.d samples from an unknown joint distribution. The BI-KSG method proposed by [5] is an improved version concerning the bias. It states that given two discrete random variables X, Y , and N , i.i.d samples $(X_1, Y_1), \dots, (X_N, Y_N)$ from the underlying joint probability distribution $f_{X, Y}(x, y)$. The BI-KSG mutual information estimator introduced in [5] is given by:

$$\begin{aligned} \hat{I}_{BI-KSG}(X; Y) &= \psi(k) + \log(N) + \log\left(\frac{c_{d_x, 2} c_{d_y, 2}}{c_{d_x, 2 + d_y, 2}}\right) \\ &\quad - \frac{1}{N} \sum_{i=1}^N (\log(n_{x, i, 2}) + \log(n_{y, i, 2})) \end{aligned} \quad (8)$$

Where:

- k is the integer that determines the k -nearest neighbour classification. Where k must be smaller than the number of samples.
- $c_{d, 2} = \frac{\pi^{d/2}}{\Gamma(\frac{d}{2} + 1)}$ is the volume of a d -dimensional unit l_2 ball.
- $\psi(k) = \frac{1}{\Gamma(k)} \frac{d\Gamma(k)}{dk}$ is the digamma function according to [5].
- $n_{x, i, p} \equiv \sum_{j \neq i} I(|X_j - X_i|_p \leq \rho_{k, i, p})$.
 - Where $n_{x, i, p}$ can be interpreted as the amount of samples within the X dimension distance of $\rho_{k, i, p}$ with respect to sample i according to [5].

D. f -CMI estimator and bound

The section presents the results from Hyarar et al. [3]. Let $R \in \mathcal{R}$ be a random variable, independent of Z , that is a source of randomness in the training of the neural network. Let's assume that $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ is a set of pairs of inputs $x \in \mathcal{X}$ and labels $y \in \mathcal{Y}$. We assume that the learning algorithm implements a function $f : \mathcal{Z}^n \times \mathcal{X} \times \mathcal{R} \rightarrow \mathcal{K}$ where \mathcal{K} is the prediction domain which can be different from \mathcal{Z} . If z is a training set, x' a test input, and r an argument that represents the stochasticity of training and predictions, then $f(z, x', r)$ is the prediction on the test example. Assume that $\tilde{Z} \sim \mu^{2n}$ is a collection of $2n$ i.i.d examples that are grouped in n pairs. The random variable $S \sim \text{Uniform}(\{0, 1\}^n)$ chooses one example from each pair \tilde{Z}_i to create the set \tilde{Z}_S of length

n . Let $L_{\text{emp}}(f, \tilde{Z}_S, R) = \frac{1}{N} \sum_{i=1}^N [l(f(\tilde{Z}_S, X_i, R), Y_i)]$ be the empirical risk of the algorithm f trained on the dataset \tilde{Z}_S with randomness R where Y_i is the corresponding label to input X_i . Similarly, let the population risk be defined as $L(f, \tilde{Z}_S, R) = \mathbb{E}_{Z' \sim \mu} [l(f(\tilde{Z}_S, X'_i, R), Y'_i)]$.

Definition III.1 (pointwise functional conditional mutual information [3]). Let u be a subset of the set $\{1, 2, 3, \dots, n\}$ of size m . Then the pointwise functional mutual information f -CMI is

$$f\text{-CMI}(f, \tilde{z}, u) = I(f(\tilde{z}_S, \tilde{x}_u, R); S_u) \quad (9)$$

and the functional mutual information is defined as

$$f\text{-CMI}_\mu(f, u) = \mathbb{E}_{\tilde{z} \sim \tilde{Z}} f\text{-CMI}(f, \tilde{z}, u) \quad (10)$$

Let $f\text{-CMI}(f, \tilde{z}, u)$ and $f\text{-CMI}_\mu(f, u)$ be written as $f\text{-CMI}(f, \tilde{z})$ respectively $f\text{-CMI}_\mu(f)$ whenever $u = \{1, 2, 3, \dots, n\}$.

Theorem III.1 (f -CMI generalization bound [3]). Let u be a random subset of size m , independent of \tilde{Z} , S and R . If $l(\hat{y}, y) \in [0, 1], \forall \hat{y} \in \mathcal{K}$ and $z \in \mathcal{Z}$, then

$$\left| \mathbb{E}_{\tilde{Z}, R, S} [L(f, \tilde{Z}_S, R) - L_{\text{emp}}(f, \tilde{Z}_S, R)] \right| \leq \mathbb{E}_{\tilde{z} \sim \tilde{Z}, u \sim U} \sqrt{\frac{2}{m} f\text{-CMI}(f, \tilde{z}, u)} \quad (11)$$

Corollary III.1.1. When $m = 1$, the bound from 11 becomes

$$\left| \mathbb{E}_{\tilde{Z}, R, S} [L(f, \tilde{Z}_S, R) - L_{\text{emp}}(f, \tilde{Z}_S, R)] \right| \leq \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{\tilde{z} \sim \tilde{Z}} \sqrt{2I(f(\tilde{z}_S, \tilde{x}_i, R); S_i)} \quad (12)$$

E. ISMI bound

This section presents the ISMI bound from [4].

Definition III.2. According to [4] the cumulant generating function of a random variable X is defined as:

$$\Lambda_X(\lambda) \triangleq \log(\mathbb{E}[e^{\lambda(X - \mathbb{E}X)}]) \quad (13)$$

Assuming $\Lambda_X(\lambda)$ exists and is convex.

Definition III.3. A convex function ψ defined in the interval $[0, b)$ where b is defined in the interval $0 \leq b \leq \infty$. Then according to [4] it's Legendre dual ψ^* can be defined as:

$$\psi^*(x) \triangleq \sup_{\lambda \in [0, b)} (\lambda x - \psi(\lambda)) \quad (14)$$

Theorem III.2. Suppose $l(\tilde{W}, \tilde{Z})$ satisfies $\Lambda_{l(\tilde{W}, \tilde{Z})}(\lambda) \leq \psi_+(\lambda)$ for $\lambda \in [0, b_+)$, and $\lambda_{l(\tilde{W}, \tilde{Z})}(\lambda) \leq \psi_-(-\lambda)$ for $\lambda \in (b_-, 0]$ under $P_{\tilde{W}, \tilde{Z}} = \mu \oplus P_W$, where $0 \leq b_+ \leq \infty$

and $-\infty \leq b_- \leq 0$. Then according to [4] the generalization error can be bounded by:

$$\text{gen}(\mu, P_{W|S}) \leq \frac{1}{N} \sum_{i=1}^n \psi_-^{*-1}(I(W; Z_i)) \quad (15)$$

$$- \text{gen}(\mu, P_{W|S}) \leq \frac{1}{N} \sum_{i=1}^n \psi_+^{*-1}(I(W; Z_i)) \quad (16)$$

Corollary III.2.1. For a logistic regression. Let

$$X \sim N(\mu_Y, \Sigma), \quad Y \in \pm 1, \quad \mu_Y \in \mathcal{R}^d$$

With a binary classifier,

$$\hat{Y} = \begin{cases} 1 & w^T X \geq 0 \\ -1 & \text{else} \end{cases}$$

With classification error $l(w, Z) = 1_{Y \neq \hat{Y}}$ and the empirical risk with n i.i.d samples being,

$$L_S(w) = \frac{1}{n} \sum_{i=1}^n 1_{\{Y_i \neq \hat{Y}_i\}}$$

Where W , the weights in the neural network of the classifier, is learnt from the loss function:

$$W = \text{argmin}_{w \in W} \frac{1}{N} \sum_{i=1}^n \log(1 + e^{-Y_i w^T X_i})$$

If $l(W, Z)$ is bounded by 1, then by Hoeffdings lemma, $l(W, Z)$ is $\frac{1}{2}$ -sub-Gaussian. Then according to [4], the ISMI bound can then be estimated by:

$$\frac{1}{n} \sum_{i=1}^n \sqrt{\frac{\hat{I}(W; Z_i)}{2}} \quad (17)$$

Where $\hat{I}(W; Z_i)$ is the the estimate of $I(W; Z_i)$.

F. Monte Carlo simulation

Monte Carlo simulation is a type of estimation that relies predominantly on the Law of large numbers. The Law of large numbers is a theorem in statistics stating that by increasing the number of i.i.d samples from a random variable, the mean of the outcomes will converge to the theoretical mean, according to [8].

IV. METHOD

In this project, we focus on creating generalization bounds, using f -CMI [3] and ISMI [4] algorithms, specifically for binary classification tasks using convolutional neural networks (CNN). The PyTorch library [9] was used for handling low-level details for machine learning-related tasks. PyTorch has optimized algorithms and functions used for many aspects of machine learning, such as optimization methods, etc, which are used in the experiments in this project. One of the datasets used in this project is the cats vs. dogs dataset from Kaggle, which contains 12500 pictures of dogs and an equal number of pictures of cats [10]. The generalization bounds must be compared to a reference generalization error, which is estimated using a Monte Carlo simulation. We explain the details for experiments for each two algorithms separately.

A. Replicating the empirical evaluation of ISMI bound for logistic regression from [4]

Some learning algorithms are difficult to analytically evaluate with the ISMI bound due to difficulty in evaluating $P_{W|Z}$. Logistical regression is one such example. This experiment aims to replicate the experiment conducted in section VI in [4]. In that particular case, they empirically estimate the ISMI bound, which we will also do in our experiment. Let $Z = (X, Y)$, where X consists of features $X \in \mathcal{R}^d$ and labels $Y \in \pm 1$. The samples generated from X are drawn from the normal multivariate distribution $X \sim N(\mu_y, \Sigma)$. Where $\mu_y \in \mathcal{R}^d$.

The algorithm to replicate the experiment can be broken down into four main phases. Creating the dataset, training the logistical regression model, calculating the estimated generalization error, and calculating the ISMI bound. The overall structure can be seen in Algorithm 1. Phases one to three are run N times to get many data points for the mutual information estimation and, therefore, the ISMI bound estimation.

In the first phase, the training dataset Z is created. It consists of $2n$ samples in pairs with a corresponding label. The samples are randomly drawn from the distributions. A validation dataset Z_2 is needed and is created in the same way as Z . For calculating the mutual information, a pair of data is randomly sampled from Z on every iteration and appended to Z_n . Only one sample is sampled per iteration as it will be paired with the sets of weights for the trained model in that iteration.

In the second phase, the logistical regression model is fitted to Z . The choice of the optimizer is significant. If we select an optimizer not dependent on the order of the samples then we only need to evaluate the bound in Corollary III.2.1 for $n = 1$ according to [4]. Stochastic gradient descent with random shuffling is one example of such an optimizer. In our implementation, we used SAG (Stochastic average gradient) with random shuffling. The weights of the trained model are appended to the variable W , the weights are W in equation 17.

The third phase is to calculate the estimate of the generalization error. It is crucial that there are enough samples in the datasets Z and Z_2 to get a stable estimate of the accuracy due to the principles brought up in Monte Carlo estimates. The estimate of the generalization error is calculated by taking the expected value of the accuracy on the training data (Z) and subtracting it from the expected value of the accuracy on the test data (Z_2).

The last phase is the calculation of the ISMI bound. The mutual information is calculated according to equation 8. The overall structure of the BI-KSG estimator implementation can be seen in Algorithm 2. The implementation closely follows equation 8. The specifics in the implementation of the steps with finding the $\log(n_{x,i,2})$ and $\log(n_{y,i,2})$ have been skipped to keep Algorithm 2 general and not programming language-specific. Readers interested in the specifics of our implemen-

tation in Python may look at our code¹. Lastly, by using the calculated mutual information and implementing equation 17 we get the estimated ISMI bound. All the steps above are repeated for every n of interest. The model parameters used for the distributions are located in table I. The values are the same as in [4], except for N , which has been changed from 5000 to 50000 to get a more stable output.

Table I
MODEL PARAMETERS

d	2
k	5
μ_1	(1, 1)
μ_2	(-1, -1)
σ	((2, 0) (0, 2))
N	50000
n	[2, 4, 6, ..., 36]

Algorithm 1 Algorithm for logistic regression

```

1: Initialization
2:  $N = 50000$ 
3:  $n_{start} = 2$ 
4:  $n_{stop} = 36$ 
5:  $n_{step} = 2$ 
6:  $\mu_1 = [-1, -1]$ 
7:  $\mu_2 = [1, 1]$ 
8:  $\sigma = [[2, 0], [0, 2]]$ 
9:  $ISMIBound = [], genError = []$ 
10:
11: for  $n$  in  $range(n_{start}, n_{stop}, n_{step})$  do
12:    $Z_n = [], W = [], estGenErrorList = []$ 
13:   for  $a$  in  $range(N)$  do
14:      $X_1 = [\text{draw } n \text{ sample from distribution 1}]$ 
15:      $X_2 = [\text{draw } n \text{ sample from distribution 2}]$ 
16:      $Y = [n \text{ zeroes}, n \text{ ones}] \{ \text{Labels} \}$ 
17:      $\{ \text{Create pairs of input and labels} \}$ 
18:      $Z = [[X_1[0], Y[0]], \dots, [X_2[N-1], Y[N-1]]]$ 
19:      $W_{model} = \text{create and train logistic regression model}$ 
    on  $Z$ 
20:      $W.append(W_{model} \text{ weights})$ 
21:      $\{ \text{Pick out a random data pair} \}$ 
22:      $Z_i.append(Z[\text{randomIndex}])$ 
23:
24:      $\{ \text{Create } Z_2 \text{ in the same way as above, needed to}$ 
    estimate gen error on the  $W$  model  $\}$ 
25:
26:      $estGenErrorList.append(\text{calculateEstGenError}(W_{model},$ 
     $Z_2, Z))$ 
27:   end for
28:
29:    $I = \text{estimate MI with BI-KSG}(W, Z_n)$ 
30:    $ISMIBound.append(\text{calculateISMIBound}(I))$ 
31:    $genError.append(\text{mean}(estGenErrorList))$ 
32:
33: end for

```

¹https://github.com/Kevin-Pettersson/ISMI_logitstic_regression

Algorithm 2 Algorithm for BI-KSG

```

1: Input:  $X, Y, k$ 
2: Output:  $\hat{I}(X; Y)$ 
3:
4: Initialization
5:  $N = \text{length of } X \text{ or } Y$ 
6: {NOTE  $X$  and  $Y$  must have same length!}
7:
8: {Calculate the volumes}
9:  $c_{d_x,2} = \text{volume of (dimension of } x\text{)-dimensional unit } 12 \text{ ball}$ 
10:  $c_{d_y,2} = \text{volume of (dimension of } y\text{)-dimensional unit } 12 \text{ ball}$ 
11:  $c_{d_y,2+d_x,2} = \text{volume of (dimension of } y + \text{dimension of } x\text{)-dimensional unit } 12 \text{ ball}$ 
12:
13: {Calculate all terms outside summation}
14:  $\text{nonSumTerms} = \Gamma(k) + \log(N) + \log\left(\frac{c_{d_x,2}c_{d_y,2}}{c_{d_x,2+d_y,2}}\right)$ 
15:
16: {Calculate summation terms}
17: for  $i$  in  $\text{range}(N)$  do
18:    $\text{sumTerms} += \frac{1}{N} \log(n_{x,i,2})$  // Summation  $x$ 
19:    $\text{sumTerms} += \frac{1}{N} \log(n_{y,i,2})$  // Summation  $y$ 
20: end for
21:  $\hat{I}(X; Y) = \text{nonSumTerms} - \text{sumTerms}$ 
22:
23: return  $\hat{I}(X; Y)$ 

```

B. Generalization bound using f -CMI algorithm

The code for this experiment is available on [github](https://github.com/rezaqorbani/f-CMI)². This repository is a fork from the repository mentioned in [3] that contains the original experiment which is reproduced and modified by us. In order to visualize the generalization bound we calculated the bound using a different number of example inputs from the dataset. Let the number of examples be denoted by n , where $n \in [75, 250, 1000, 4000]$. For each n , k_1 samples \tilde{z} of \tilde{Z} with length $2n$ are selected. These samples are grouped in n pairs and are each split into k_2 different training/test splits. The way these splits are created is by randomly selecting the indices s from $S \sim \{0, 1\}^n$ with fixed seed $m \in \{1, 2, \dots, k_1\}$. The reason the seeds are fixed is to make the results the same each time they are reproduced. Selecting n examples from the set \tilde{z} according to pair indices s creates \tilde{z}_S . These examples are selected by randomly selecting n indices, which choose which example in each pair is chosen, with a fixed seed. The indices are then saved as they are later used for calculating the conditional mutual information as described below. The CNN for binary classification is created using the configurations in II and it is then trained with the configuration described in III.

1) *Generalization error:* In order to compare the estimated generalization bound using the experiments above to the real generalization error, we need to estimate the generalization error. The generalization error cannot be calculated directly and thus has to be estimated. The estimation is calculated in several

parts. First $L_\mu(f, \tilde{Z}_S, R) - L_S(f, \tilde{Z}_S, R)$ is estimated by taking the average error – that is the number of correctly predicted input divided by all total number of inputs – over the training examples minus the average error over the test examples. Then to get an estimate of $\hat{g}(\tilde{z}) \triangleq \mathbb{E}_{S,R}[L_\mu(W) - L_S(W)]$ we average over $k_2 = 30$ samples of S and R . It remains then to take the expectation over \tilde{z} . By averaging $g(\tilde{z})$ over $k_1 = 5$ samples of \tilde{Z} both the expected value and the standard deviation for the generalization error can be calculated.

2) *calculating mutual information:* The mutual information is calculated between the predictions and the randomly selected indices according to equation 4 and theorem III.1. The estimation of $f\text{-CMI}(f, \tilde{z}, \{u\}) = I(f(\tilde{z}_S, \tilde{x}_u, R); S_u)$ where $u = [1 \ 2 \ \dots \ u] \in \{1 \ 2 \ \dots \ n\}$ is done in a similar manner as above by averaging over k_2 samples of S and R . Then, averaging $f\text{-CMI}_\mu(f, u) = \mathbb{E}_{\tilde{z} \sim \tilde{Z}} f\text{-CMI}(f, \tilde{z}, u)$ over k_1 samples of \tilde{Z} gives the conditional mutual information.

Table II
PARAMETERS FOR CNN USED IN THE f -CMI EXPERIMENT

Layers	Properties
Convolutional	32 Filters, 4×4 kernels, stride 2, padding 1, batch normalization, ReLU
Convolutional	32 Filters, 4×4 kernels, stride 2, padding 1, batch normalization, ReLU
Fully connected	128 units, ReLU
Fully connected	2 units, linear activation

Table III
TRAINING CONFIGURATION FOR BINARY CLASSIFICATION TASK IN THE f -CMI EXPERIMENT

Optimizer	ADAM with 0.001 learning rate and $\beta = 0.09$
Number of examples (n)	[75, 250, 1000, 4000]
Number of epochs	200
Number of samples for \tilde{Z} (k_1)	5
Number of samplings for S for each \tilde{z} (k_2)	30

V. RESULTS

In this section, we separately present the results for each experiment described above.

A. Replicating empirical evaluation of ISMI bound for logistic regression [4]

The results from our replication of the experiment are in Fig. 1. The overall result indicates that our implementation yields a result closely matching the original authors, except for $n = 2$ and $n = 4$. At the specified n 's, our bound is not as tight. Our implementation also seems to yield a less stable output, even with the number of samples (N) increased to 50000. The original author's generalization error was left out of the graph since our results were more or less identical.

²<https://github.com/rezaqorbani/f-CMI>

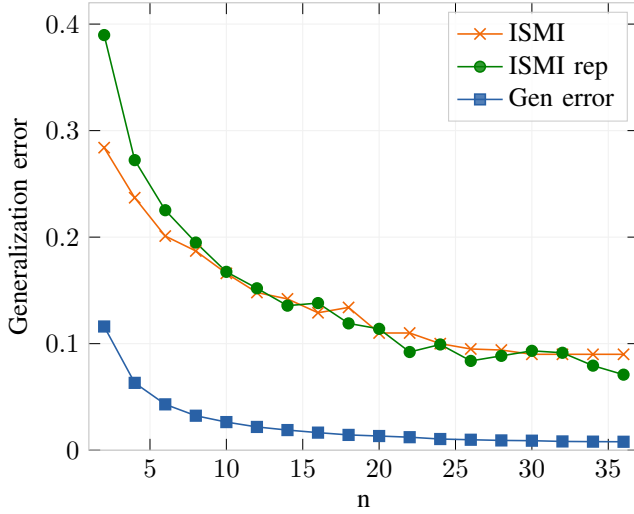


Fig. 1. Results from replicating the empirical evaluation of the ISMI bound for logistical regression [4]. ISMI: The original ISMI bound result from [4]. ISMI rep: Our ISMI bound result. Gen error: The estimated generalization error.

B. f -CMI bound experiment [3]

Generalization bounds for binary classification of digits 4 vs 9 from the MNIST dataset can be found in [3]. We apply the same algorithm on the binary classification of cats and dogs from the dataset found on [10]. Although the algorithm for calculating the generalization bound is the same, the dataset, the architecture of the neural network, and the number of epochs have been altered in our experiment but the different number of examples for which the generalization bounds are calculated is the same. The results are presented in Fig. 2. We can clearly see that the calculated generalization bound bounds the generalization error to give an upper bound.

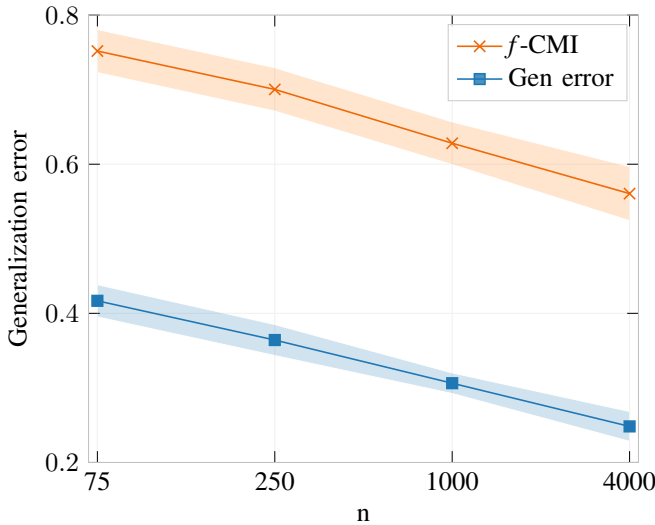


Fig. 2. Calculating generalization bound of binary classification of cats and dogs using the f -CMI algorithm

VI. DISCUSSION

A. ISMI bound

The results from the replication of the empirical evaluation of the ISMI bound for logistic regression from [4] yielded very similar results to the original authors [4]. We see that the ISMI bound closely follows the decreasing trend from the estimated generalization error. This result is expected because the greater number of input examples makes the algorithm better at predicting labels for the input. The results also show that the bound is moderately greater than the true generalization error. However, as previously mentioned, our results yield less stable output when $n \geq 14$, and we are uncertain as to the reason why. It is also unclear why this would be the case, specifically since we increased the number of trials per number of examples, N to 50000 from 5000. We hypothesize that the authors in [4] have made a mistake in the writing and that the number of data points used was considerably larger. There is also the anomaly with the $n \leq 4$. In our experiment, it gives a result much greater than the original authors. We think it could be related to the randomness in the generation of data, that when n is small, the randomness plays a bigger role in the result. Since no randomness seed was given by the authors, implicating it would be impossible to get the same data. Future work might explore the reasons why our results are less stable and why they differ considerably for $n \leq 4$.

B. f -CMI bound

The result from our experiment shows a more loose bound than the one found in [3]. This could, to some extent, be because the classification of cats and dogs where each image in the dataset [10] has a greater number of pixels (400×300 pixels or more) compared to images in the MNIST dataset [11] (28×28). Also, the pictures in our dataset are much more complex since they are taken from the real world where there are objects other than the one we are classifying present in the picture. This makes both the task of classification and calculation of the generalization bound more complex. This is evident in the fact that although the number of epochs for training was decreased to 20 in our experiment from 200 in [3], the combined time for the experiment increased nearly 2.5 times. As in the experiments in [3] we see that the generalization bound and the generalization error decrease with the increasing number of examples, n . The reason for this is that the higher number of input examples makes the algorithm better at predicting labels of the respective input. Changing the number of epochs from 200 to 20 could perhaps make the bound more loose compared to the results in [3], however, we were unable to run the training with 200 epochs because it would take an unreasonably long time. Another factor for the somewhat loose bound from our experiment could be that we have altered the CNN to have less layers and filters.

VII. CONCLUSION

In this project, we explored two different methods for bounding generalization error. We introduced the information-theoretic concepts of entropy and mutual information. Using

these concepts, we then described two recent algorithms, f -CMI and ISMI, for estimating the generalization error bounds and two mutual information estimators, BI-KSG and f -CMI. After reproducing the experiments from the mentioned papers with few modifications, we were able to produce results that seem to agree with the original experiments from which they were reproduced.

ACKNOWLEDGMENT

The authors would like to thank Amaury Gouverneur for all his assistance and guidance in the project!

REFERENCES

- [1] J. Jumper *et al.*, “Highly accurate protein structure prediction with alphafold,” *Nature*, vol. 596, p. 583–589, Jul. 2021. [Online]. Available: <https://www.nature.com/articles/s41586-021-03819-2>
- [2] L. Hu and Y. Chen, “Fair classification and social welfare,” in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, ser. FAT* ’20. New York: Association for Computing Machinery, Jan. 2020, p. 535–545. [Online]. Available: <https://doi.org/10.1145/3351095.3372857>
- [3] H. Harutyunyan, M. Raginsky, G. Ver Steeg, and A. Galstyan, “Information-theoretic generalization bounds for black-box learning algorithms,” in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Red Hook: Curran Associates, Inc., 2021, pp. 24 670–24 682. [Online]. Available: <https://proceedings.neurips.cc/paper/2021/file/cf0d02ec99e61a64137b8a2c3b03e030-Paper.pdf>
- [4] Y. Bu, S. Zou, and V. V. Veeravalli, “Tightening mutual information-based bounds on generalization error,” *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 121–130, may 2020. [Online]. Available: <https://doi.org/10.1109/JSAIT.2020.2991139>
- [5] W. Gao, S. Oh, and P. Viswanath, “Demystifying fixed k-nearest neighbor information estimators,” *IEEE Transactions on Information Theory*, vol. 64, no. 8, pp. 5629–5661, Feb. 2018.
- [6] T. M. Cover and J. A. Thomas, *Elements of information theory*, 2nd ed. New York: Wiley-Interscience, 2006.
- [7] A. Kraskov, H. Stögbauer, and P. Grassberger, “Estimating mutual information,” *Phys. Rev. E*, vol. 69, p. 066138, Jun. 2004. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevE.69.066138>
- [8] (2022, Apr) Law of large numbers. [Online]. Available: <https://www.britannica.com/science/law-of-large-numbers>
- [9] A. Paszke *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, Eds. Red Hook: Curran Associates, Inc., Dec. 2019, pp. 8024–8035. [Online]. Available: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
- [10] J. Elson, J. J. Douceur, J. Howell, and J. Saul, “Asirra: A captcha that exploits interest-aligned manual image categorization,” in *Proceedings of 14th ACM Conference on Computer and Communications Security (CCS)*. Association for Computing Machinery, Inc., Oct. 2007. [Online]. Available: <https://www.microsoft.com/en-us/research/publication/asirra-a-captcha-that-exploits-interest-aligned-manual-image-categorization/>
- [11] L. Deng, “The mnist database of handwritten digit images for machine learning research,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, Nov. 2012.

Experiments of Federated Learning on Raspberry Pi Boards

Farhad Madadzade and Simon SONDÉN

Abstract—In recent years, companies of all sizes have become increasingly dependent on customer user data and processing it using machine learning (ML) methods. These methods do, however, require the raw user data to be stored locally on a server or cloud service, raising privacy concerns. Hence, the purpose of this paper is to analyze a new alternative ML method, called federated learning (FL). FL allows the data to remain on each respective device while still being able to create a global model by averaging local models on each client device. The analysis in this report is based on two different types of simulations. The first is simulations in a virtual environment where a larger number of devices can be included, while the second is simulations on a physical testbed of Raspberry Pi (RPI) single-board computers. Different parameters are changed and altered to find the optimal performance, accuracy, and loss of computations in each case. The results of all simulations show that fewer clients and more training epochs increase the accuracy when using independent and identically distributed (IID) data. However, when using non-IID data, the accuracy is not dependent on the number of epochs, and it becomes chaotic when decreasing the number of clients which are sampled each round. Furthermore, the tests on the RPIs show results which agree with the virtual simulation.

Sammanfattning—På den senaste tiden har företag blivit allt mer beroende av kunders användardata och har börjat använda maskininlärningsmodeller för att processera datan. För att skapa dessa modeller behövs att användardata lagras lokalt på en server eller en molntjänst, vilket kan leda till integritetsproblematik. Syftet med denna rapport är därför att analysera en ny alternativ metod, vid namn "federated learning" (FL). Denna metod möjliggör skapandet av en global modell samtidigt som användardata förblir kvar på varje klients enhet. Detta görs genom att den globala modellen bestäms genom att beräkna medelvärdet av samtliga enheters lokala modeller. Analysen av metoden görs baserat på två olika typer av simuleringar. Den första görs i en virtuell miljö för att kunna inkludera större mängder klientenheter medan den andra typen görs på en fysisk testbänks som består av enkortsdatorerna Raspberry Pi (RPI). Olika parametrar justeras och ändras för att finna modellens optimala prestanda och noggrannhet. Resultaten av simuleringarna visar att färre klienter och flera träningsperioder ökar noggrannheten när oberoende och likafördelad (på engelska förkortat till IID) data används. Däremot påvisas att noggrannheten inte är beroende av antalet perioder när icke-IID data nyttjas. Noggrannheten blir däremot kaotisk när antalet klienter som används för att träna på varje runda minskas. Utöver observeras det även att testresultaten från RPI enheterna stämmer överens med resultatet från simuleringarna.

Index Terms—Federated Learning, Raspberry Pi, FedAvg, Decentralized, Machine Learning, Convolutional Neural Network, PyTorch.

Supervisors: Ming Xiao and Hao Chen

TRITA-number: TRITA-EECS-EX-2022:172

I. INTRODUCTION

Machine learning (ML) is used to utilize data that have been gathered to create models that predict outcomes without the need for hard-coding instructions for such events. The concept is often used by companies to create practical software that can be used for, for instance, natural language processing, computer vision, and speech recognition, among others. Though most people are only aware of its use in consumer products, such as smartphones, ML is also used in data-intensive fields, such as the medical industry, to better analyze experimental data. Nonetheless, their commonality is that great amounts of data are often handled. In some cases, this data can be private user data or even highly confidential data.

Cyber security has become an increasingly relevant topic after the number of cyber-attacks has increased over the past few years, as noted in [1]. According to [2], data breaches, where an attacker gains access to a company's private records, have also increased in prevalence. The information is then often released to the public or the highest bidder. A recent example, published in many newspapers, such as the BBC [3], is of a leak where 540 million Facebook user records were exposed openly. To combat this issue, several different approaches can be taken to increase the security of the server-stored data. It would, however, be more optimal if the private data would never leave the users' devices. This is possible using a decentralized approach, i.e., federated learning (FL).

FL, as first described in [4], is an ML method where the objective is to train a qualitative centralized model without the need for any training data on a local server. Instead, all raw training data remains on the client devices, often represented by mobile phones. In this constructed environment, each client device uses a learning algorithm that creates a model based on the local data. Furthermore, the server receives the models from all its clients and proceeds to average these into a new global model. This process replaces the conventional ML methods where raw training data are used to create a model on the server.

Decentralized ML, e.g., FL, can handle the aforementioned concerns. It does, however, come with its challenges in other areas which are not issues in traditional methods. This is mainly concerning the fact that the FL process is dependent on good network connections between the client and server, which cannot always be guaranteed. This would result in some local models not being accessible. This bachelor thesis, therefore, aims to simulate FL on both virtual and RPI clients to be able to evaluate both the performance of the FL implementation, but also to promote communication efficiency.

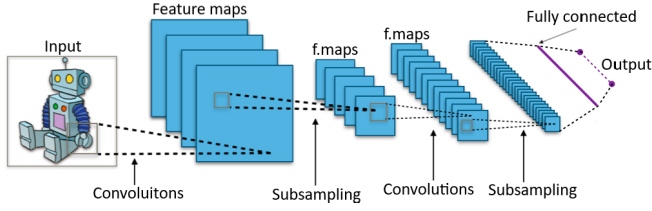


Fig. 1. Typical CNN architecture. Adapted and modified from [6].

II. BACKGROUND

The topic of FL includes many concepts whose background information is necessary to understand the methods and results presented. This section will therefore explain such concepts.

A. Artificial Neural Network

As described in [5], artificial neural networks (ANN) are networks used to create prediction models. These networks are based on biological nervous systems, hence the name. ANNs are comprised of layers that consist of "interconnected computational nodes", also called neurons. These work together to learn and ultimately make predictions based on the input data. This data is entered into the input layer as a multidimensional vector and later transferred to one or more hidden layers. A hidden layer is responsible for making choices based on the previous layer and it considers how a stochastic adjustment either makes the predictions better or worse. Collectively, this is called "learning" and is often done using stochastic gradient descent (SGD) leveraging backpropagation. When multiple hidden layers are put together, one achieves what is referred to as deep learning. The final layer in an ANN is the output layer, which, as the name states, is responsible for outputting the final result.

B. Convolutional Neural Network

A convolutional neural network (CNN) is one typical architecture of an ANN. A typical CNN architecture can be observed in Fig. 1, showing the input, hidden, and output layer. As described in [5], the most noteworthy difference between a regular ANN and a CNN is that CNNs are often used for image-based pattern recognition. In the process, due to it doing a fixed size convolution, this results in a reduction in parameters in comparison to ANNs, where each neuron is connected to all other neurons in the next layer.

C. Federated Learning Process

FL is an approach to ML which differs from the traditional setting. Traditionally, the training data from the client devices, which are needed for the learning, have to be collected and accessible to a central server. Hence, the data must be stored locally or on a cloud-like service, and then used for training centrally. With FL, the user devices are instead used as nodes for computations. Each device creates its local model which is then sent to the other models to be combined. This can be done in two different ways. The first is centralized, where all models are sent to a central server where they are then combined to

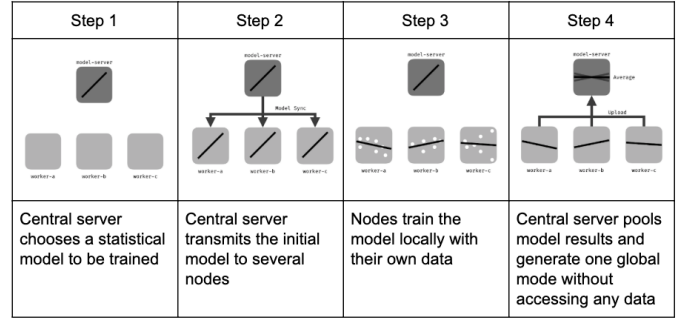


Fig. 2. Federated learning general process in central orchestrator setup. Adapted from [7].

make a global model. The second way is decentralized, where each client sends its model to all other clients which are then combined on each device. Both of these allow client devices to "collaboratively learn a shared prediction model", as stated in [4]. The whole process is also illustrated in Fig. 2.

D. Stochastic Gradient Descent Method

According to [8], SGD is an iterative process often used in ML as an optimization method. In this context, the objective is to minimize the total loss $Q(w(t))$, which can be denoted as

$$Q(w(t)) = \frac{1}{n} \sum_{i=1}^n Q_i(w(t)), \quad (1)$$

where w is a vector of the tunable parameters of the model and $Q_i(w(t))$ is the value of the loss function at the i -th observation in the dataset of size n . The task is therefore to estimate the value of the parameter $w(t)$ at which this minima is achieved. The SGD algorithm then uses the gradient of $Q(w(t))$ to update a new estimate of $w(t)$ iteratively. The new estimate is given by

$$w(t+1) = w(t) - \eta \nabla Q(w(t)), \quad (2)$$

where η is the step size (which is often called the learning rate when working with neural networks). The function is stochastic as it estimates the gradient, where the estimation is based on a randomly selected fraction of the data.

E. Federated Averaging Algorithm

The federated Averaging (*FedAvg*) strategy for combining different models was proposed in the seminal paper [4]. It works by averaging the weight of all the models. According to [9], it is essentially based on another algorithm called federated SGD (*FedSGD*). This algorithm takes a fraction of the involved clients, denoted by C , and computes the gradient of the loss of all the data of these clients. The *FedAvg* algorithm is a special case of *FedSGD* where $C = 1$ and the learning rate is η . The algorithm makes each client k compute an averaged gradient g_k on its local data with the current global model w_t . The server then receives all of these gradients and computes the next iteration of the model with

$$w_{t+1} \leftarrow w_t - \eta \sum_{k=1}^K \frac{n_k}{n} g_k, \quad (3)$$



Fig. 3. A few samples from the MNIST test dataset. Adapted from [13].

where K is the number of clients, n_k is the partition size of client k 's data, and n is the total number of data points.

F. PyTorch

This project will make use of an ML library called *PyTorch*. According to [10], PyTorch is an open-source project which is integrated into the *Python* ecosystem. It was developed by Facebook's (now known as Meta) artificial intelligence research lab, FAIR. Its purpose is to make debugging easier while maintaining efficiency with the help of graphical processing unit (GPU) acceleration support.

G. Flower

Flower is an FL framework intended to simplify the implementation of centralized FL. The framework specializes in larger-scale experiments and it is stated in [11] that experiments have shown that Flower can handle up to 15 million clients using only two high-end GPUs. This allows researchers to easily transfer simulations of FL to an environment containing real devices to continue further studies. Flower also supports all big frameworks such as TensorFlow and PyTorch, making ML easier to implement.

H. Dataset

1) *MNIST*: First appearing in [12], the Modified National Institute of Standards and Technology, more commonly known as MNIST, is a collection of handwritten digit (0-9) images, see Fig. 3. The MNIST database has become the standard for testing ML algorithms for image and pattern recognition purposes. There are a total of 60,000 images that can be used for training, and a total of 10,000 testing images. The two types are both created from the same distribution. The images are colored black and white (black digits, white backgrounds) and have dimensions of 28-by-28 pixels. Therefore, the image vector, which is to be used in the CNN, will have 784 binary elements.

2) *Independent and Identically Distributed Random Data*: Written in [14], independent and identically distributed random variables (in this setting the variables are data) are variables that satisfy two conditions. The first condition is

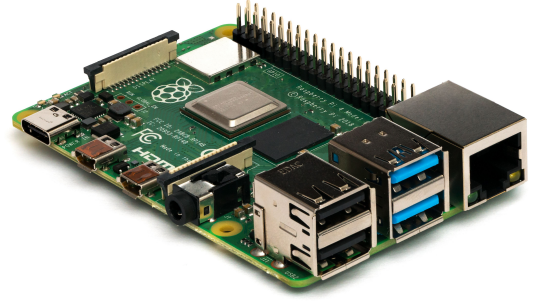


Fig. 4. Raspberry Pi 4 Model B from the side. Adapted from [18].

that all the variables in question are independent of each other. This means that the characteristics of one variable are not dependent on the previous variable. The second condition is that they are identically distributed, implying that all the variables are generated from the same probability distribution and that there is no general trend. *Non-IID* is, naturally, when these conditions, one or both of them, are not satisfied.

I. Docker

Docker containers are used in this project to allow for reproducible builds and to make the algorithms work on different computers. According to its documentation in [15], a container is software that can be used to unpack code and any necessary files. This allows for the applications to run more easily and efficiently in any computing setting. In this project, docker containers allowed the client and server-side, as well as all the virtual simulations, to be executed and assessed efficiently.

III. MATERIAL

This report only utilizes one element of material to finalize the results and simulations. This material consists of a series of single-board computers called Raspberry Pi (RPI), manufactured by the Raspberry Pi Foundation. According to [16], the original RPI was released in 2012. This model had a single-core 700 MHz central processing unit (CPU) and 256 MB of RAM. The testbed of RPIs used in this report will use the latest model shown in Fig. 4, the Raspberry Pi 4 Model B. As specified in [17], this model comes equipped with a 1.5 GHz CPU and 8 GB of RAM. The CPU is a 64-bit quad-core ARM Cortex-A72 system on chip (SoC). Hence, the graphical processing unit (GPU) is included in the main chip of the computer. Because there is not a dedicated GPU, the overall processing power of the RPIs in the testbed will have worse performance than most phones in real life. This is a factor that will be accounted for in the method in subsection IV-B to allow the simulations to run at a better rate.

IV. METHOD

This project's methodology includes various methods that are needed to achieve the final result. Some are applied to all the simulations of the project, and some are used for specific parts of the simulations. All parameter values can be found in the GitHub repository found in Appendix A.

A. Organization of the Dataset

In a real-world scenario, each client would generate their training data on their device. This leads to the generated dataset potentially being distributed in a non-IID way since different clients can be more likely to generate a specific type of data. When training a neural network there is an IID assumption, without that assumption the performance might get worse. Both scenarios will therefore be tested.

Since the simulations will be done using the MNIST dataset described in subsection II-H1, the dataset has to be distributed to all clients before the training can begin. This can be done in the two different ways described below.

1) *IID Data*: IID data can be created by randomizing the whole dataset and then distributing it to the clients as desired. In this case, it is done by either keeping the total amount of data constant, or the data per client constant.

2) *Non-IID Data*: To generate non-IID data as described in subsection II-H2, the data is organized in a manner such that the images from the original dataset are sorted in an ascending order based on the number they represent. When the clients are set to train their models, the dataset is divided into equal segments for each client. This means that each client only gets some specific digits, making the dataset not identically distributed. Data distributed this way can also be called a heterogeneous distribution since it is not homogeneous and different clients get different amounts of some types of data.

B. Configuration of the Neural Network

For the training routine, PyTorch was used to create and configure the neural network. This was done using the `torch.nn` module from PyTorch, which is used to configure neural networks. For the simulations which are to be presented in this report, CNNs were implemented but not used. Instead, a fully connected network with one hidden layer is used. These are created using built-on `Linear` layers, which, as described in the documentation in [19], apply a linear transformation to the input data. The reason for this is that the RPIs do not have dedicated GPUs, as opposed to the smartphones which they represent, which makes it more difficult to generate results from the simulations. Another method, `Dropout`, is also used. This method zeros out some elements of the input tensor with a specified probability. In the documentation in [20], it is noted that this is a good way of preventing co-adaption of the neurons and decreasing the risk of over-fitting.

To be able to approximate non-linear functions, Rectified Linear Unit (ReLU) was used as the activation function between layers. When training the SGD optimizer `optim.Adam` (from the PyTorch module) was used with a starting learning rate of 0.01. To further increase performance, step-wise decrements were made to the learning rate each epoch. The decrement size was varied depending on the data distribution; a multiplication by 0.7 was used when using IID-data, and 0.15 was used when working with non-IID data.

C. Centralized Evaluation

Centralized evaluation is possible to perform using the already built-in functionality in the *Flower* framework which

was implemented. As described in the documentation for centralized evaluation in [21], the main idea is to add another parameter, `eval_fn`, to the *strategy* which is to be used in the FedAvg algorithm. This parameter is a function that is specified and can be found in the source code of *simulation* through the link in Appendix A. This addition to the algorithm allows for centralized evaluation, where the evaluation is made on the global model created with the client models. This is an alternative to client-based evaluation where local models are evaluated on the local models, and whose result is later sent to the server for further use. By doing the centralized evaluation, less communication is needed, which promotes communication efficiency. The strategy also includes a parameter called `fraction_fit` which decides what fraction of the clients is to be sampled for training and ultimately generating the global model. This addition allows for all clients to receive a global update without the need for all clients to be available for communication with the server.

D. Program Design

The work process of creating the required programs to be able to execute the simulations was initiated by creating the necessary docker files. These docker files are used to easily build and run the needed programs for the simulation. One docker file was created for the server, one for the virtual clients, and one for the RPI clients. They have separate files because they have different configuration necessities, such as clients needing an identification number to access the correct training data.

The essential programs, which can be found in the Github repository linked in Appendix A, are `client`, `mnist`, `server`, and `simulation`. The programs `server` and `client` are responsible for starting the server and clients, respectively. The `mnist` program contains the necessary code to configure the neural network as described in subsection IV-B, data partitioning and loading, and the training and testing of data. These were implemented using both PyTorch and the Flower framework, where the Flower framework was responsible for implementing FedAvg. The `simulation` is used to start the virtual simulations. It contains functions responsible for data partitioning and the functions to initiate `client` and `server`. This program is specifically used when performing the virtual simulations and allows different parameters, such as the fraction of clients to be chosen for training each round, to be changed as needed. When performing simulations on the physical testbed of RPIs, `simulation` is not needed.

V. RESULTS

The results will be from both simulated results and the testbed established on the RPIs.

A. Virtual Simulations

The series of tests to be presented were made using a virtual simulation which allowed for large amounts of client devices to be included in the simulation. The results were generated by using 10, 30, and 100 clients respectively.

1) *Varying the Number of Clients Using IID Constant Total Data:* Fig. 5 shows the results of varying the number of clients on centralized accuracy and loss where the total amount of data is constant and is distributed amongst clients in an IID way, as described in subsection II-H2.

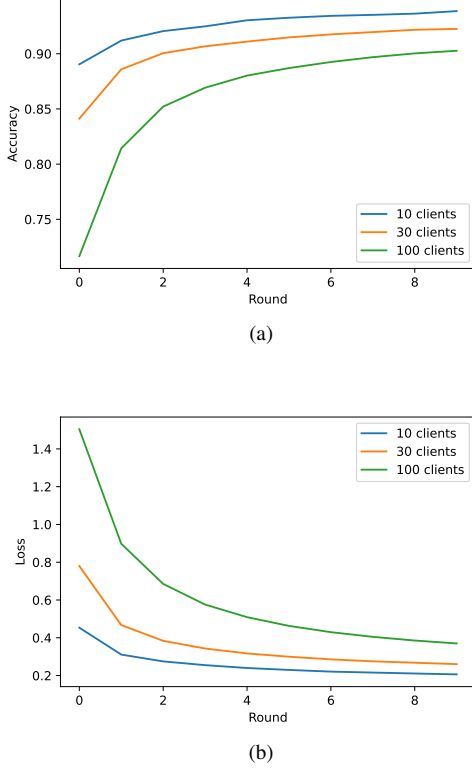


Fig. 5. Virtual simulation with centralised evaluation on test data, training on IID data and varying the total number of clients: (a) accuracy; (b) loss.

2) *Varying the Number of Clients Using non-IID Constant Total Data:* Fig. 6 shows the results of varying the numbers of clients on centralized accuracy and loss where the total amount of data is constant and is distributed amongst clients in a non-IID way, as described in subsection IV-A2.

3) *Varying the Number of Clients Using IID and non-IID Constant Client Data:* Fig. 7 shows the results of varying the numbers of clients on centralized accuracy on both IID and non-IID data where each client holds the same amount of data regardless of the number of clients used.

4) *Varying the Number of Local Epochs Using IID Data:* Fig. 8 illustrates the results of varying the total number of local epochs on centralized accuracy and loss, where the data is distributed in the IID way described in subsection II-H2.

5) *Varying the Number of Local Epochs Using non-IID Data:* In Fig. 9, the results are given by varying the total number of local epochs on centralized accuracy and loss where the data is distributed in a non-IID way described in subsection IV-A2.

6) *Varying the Number of Clients to Train on Each Round Using IID Data:* Fig. 10 shows the results of varying the number of clients to train on each round, on centralized accuracy and loss, where the data is distributed in an IID way described in subsection II-H2.

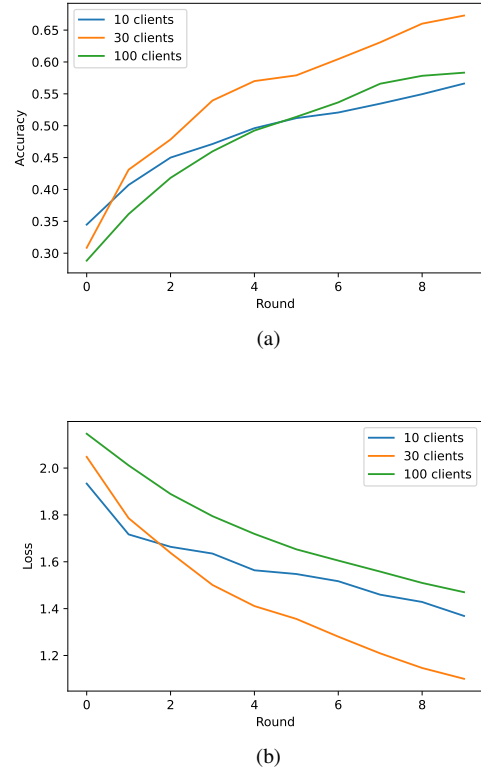


Fig. 6. Virtual simulation with centralised evaluation on test data, training on non-IID data and varying the total number of clients: (a) accuracy; (b) loss

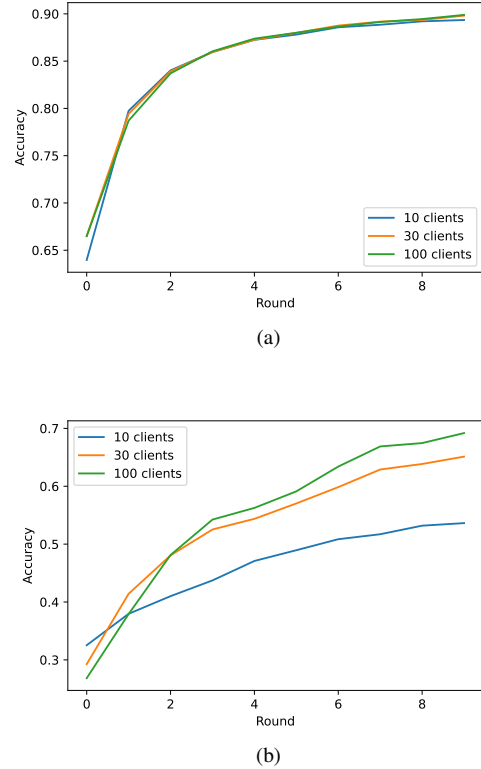


Fig. 7. Virtual simulation with centralised evaluation on test data, varying the total number of clients with constant amount of data: (a) IID; (b) non-IID

7) *Varying the Number Clients to Train on Each Round using non-IID Data:* Fig. 11 demonstrates the effect of varying

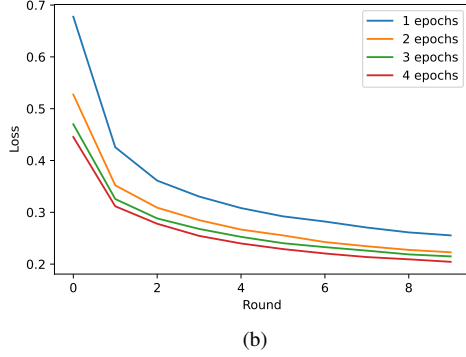
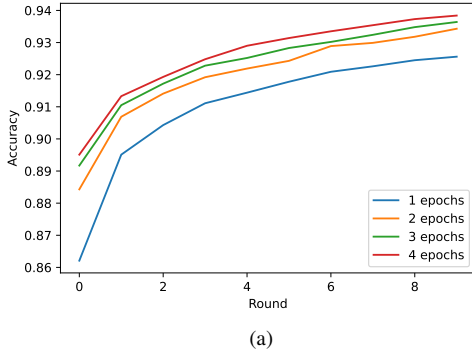


Fig. 8. Virtual simulation with centralised evaluation on test data using IID data, varying the number of local epochs: (a) accuracy; (b) loss.

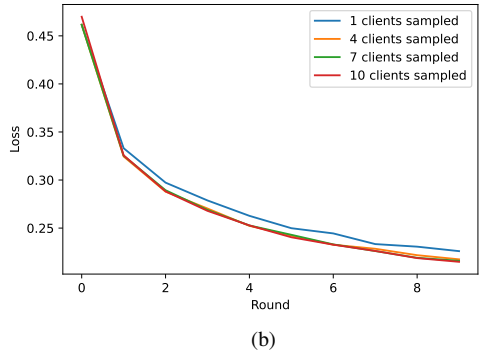
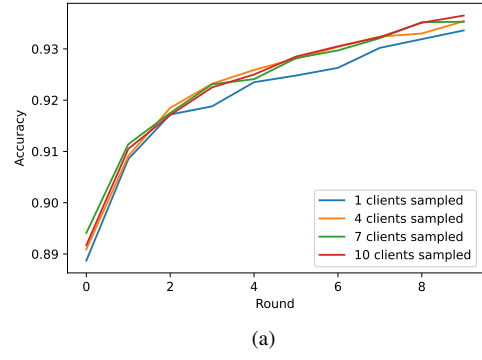


Fig. 10. Virtual simulation with centralised evaluation on test data using IID data, varying the number of clients sampled out of total amount (10 clients) to randomly train on: (a) accuracy; (b) loss.

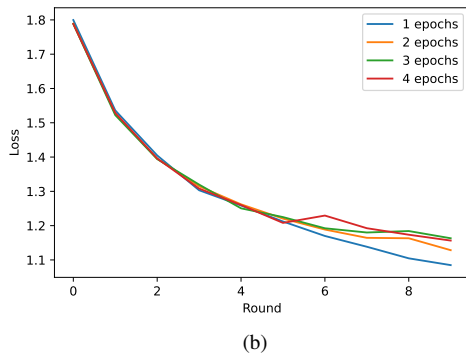
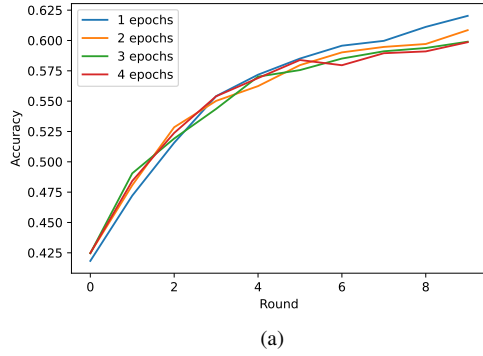


Fig. 9. Virtual simulation with centralised evaluation on test data using non-IID data, varying the number of local epochs: (a) accuracy; (b) loss.

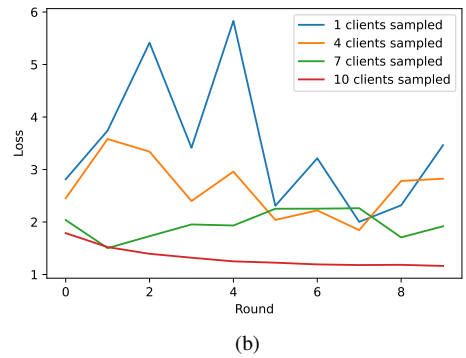
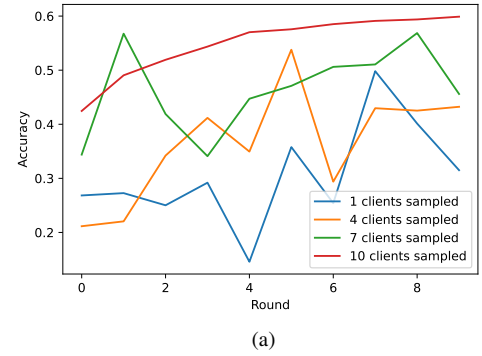
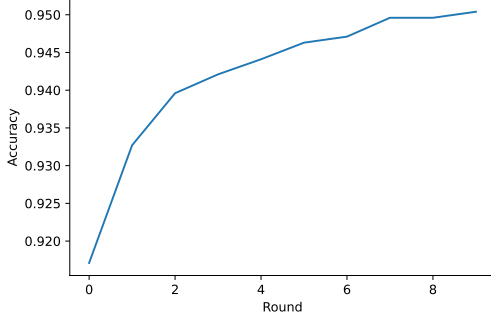


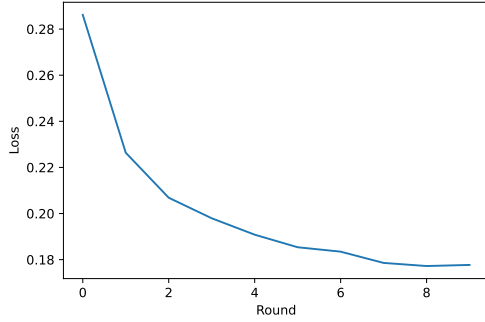
Fig. 11. Virtual simulation with centralised evaluation on test data using non-IID data, varying the number of clients out of total amount (10 clients) to randomly train on: (a) accuracy; (b) loss.

the number of clients to train on each round, on centralized accuracy and loss, where the data is distributed in a non-IID

way described in subsection IV-A2.



(a)



(b)

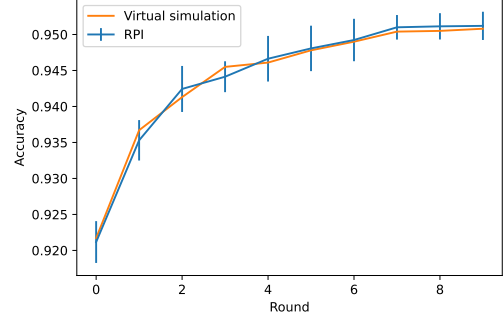
Fig. 12. Centralized evaluation on test data using IID data on two raspberry pi:s each functioning as separate clients: (a) accuracy; (b) loss.

B. RPI Simulations

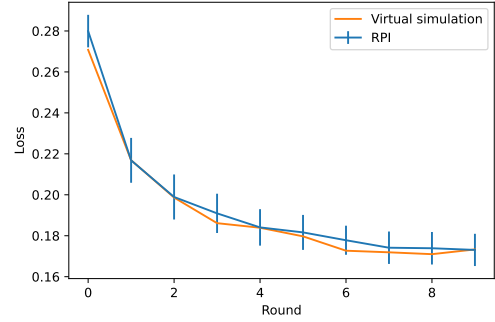
The results of the accuracy and loss over 10 rounds using the testbed consisting of two RPIs as clients can be seen in Fig. 12. The tests were made using IID data. As can be noted, these plots have similar results as those seen when using 10 clients in Fig. 5.

C. Comparison Between RPI and Virtual Clients

The results of accuracy and loss over 10 rounds using two RPIs and two virtual clients, respectively, can be seen in Fig. 13. The plots show similar characteristics in the accuracy and loss.



(a)



(b)

Fig. 13. Centralized evaluation on test data using IID data, comparing the results using two RPI:s with two virtual clients: (a) accuracy; (b) loss.

need to learn how to differentiate between digits. Instead, it can cheat by only giving the same output each turn regardless of input since the correct output label is always the same.

In Fig. 7 (a), where each client receives the same amount of data regardless of the total number of clients, a small difference in performance for the IID case is found. This is probably due to the MNIST data being very similar for each class, meaning only a few samples from each are needed to generalize. In the non-IID case, using more clients, which results in more total data, is necessary for better accuracy. This effect can be seen in Fig. 7 (b), where the accuracy when implementing 100 clients is greater than that of 30 clients, as opposed to the case in Fig. 6 where the amount of data is not constant.

By varying the number of epochs each client trains on the IID data before sending it back to the server, Fig. 8 is given. There it is noted, as expected, that increasing the number of epochs increases the final accuracy. This is due to the fact each client model has more time to fit and converge closer to an optimum. For the non-IID case, as seen in Fig. 9, there is less of a dependence on the number of epochs. This is probably due to the same fact as hinted earlier, that when using non-IID data there is not as much training needed to be done since each client only trains on one specific digit. This means increasing the number of epochs can even have a negative effect since it means that each model becomes over-fitted on the client's specific digit.

In Fig. 10, where the amount clients used to train on is

VI. DISCUSSION

In Fig. 5, it is clear to see that decreasing the number of clients increases the overall performance with the highest accuracy of 94% achieved by using 10 clients. This is to be expected since each client receives more data and we average over fewer models which leads both to better individual models and global models. This is, however, not as clear in the case when the same experiment is done using non-IID data, as seen in Fig. 6. Here, a more chaotic relationship is seen, where using 30 clients has the highest accuracy of 66% after 10 rounds of training. It should also be noted that its highest accuracy is still a lot lower than the highest accuracy obtained when using IID data. This is because each model will mostly only see one type of digit and therefore does not

varied, it is noted that the number of clients sampled does not matter as much when working with IID data. The only difference that can be seen is that using only one client is slightly worse than using more. For the non-IID, the story is a lot different, as seen in Fig. 11. There, the accuracy and loss are getting unstable when using fewer than the total number of clients. This is due to each client having mostly different digits from the other clients. Depending on the clients, randomly selected new digits might be learned and others might be forgotten which would ultimately result in high variance.

As depicted in Fig. 12, we can see that the testbed is working as intended and yields a performance of 95% accuracy. This is similar to the results in Fig. 5 where 10 clients are used. The slightly higher performance follows the trend described above where fewer clients are used which yields higher results. Fig. 13 also shows similarities between simulating two clients virtually and using two RPIs, indicating that the results from the RPIs should be concurrent with the virtual simulations, even if more RPIs were to be included.

As described in the methodology of this bachelor thesis, two different measures can be considered for promoting communication efficiency. The first was to implement the centralized evaluation. Centralized evaluation does not require any further communication since the evaluation is done on the server. The evaluation results from the alternative, decentralized evaluation, would, however, need to be communicated to the server if it is needed for further use. The second measure to promote communication efficiency is to sample fractions of the clients for training, as all clients might not always be available. The results in Fig. 10 show that the accuracy is not substantially affected by sampling a smaller fraction of clients for training in the IID case. Although the MNIST dataset is fairly easy to achieve good results with, this shows that not all clients need to be used to create a good generalization. On the other hand, the same results from the non-IID case, seen in figure 11, were not as conclusive, as the results were more chaotic, and would therefore require further improvements. Nevertheless, it is possible to use these means to solve the shortcomings of FL such that it can be used to solve the privacy issues of traditional ML methods.

VII. CONCLUSION

We were able to create software to achieve FL on a testbed as well as create software for virtual simulation which allowed for a greater amount of clients to be included. The study in the virtual environment concludes that the results were as expected when using IID data; the accuracy increased (and loss decreased) when the number of total clients, rounds, and epochs increased. However, some unexpected results were observed when using non-IID data. The effect of varying the total number of epochs did not have a significant effect. Nonetheless, varying the total number of clients showed that using 30 clients resulted in higher accuracy than using 100, as opposed to the IID case. By varying the number of clients randomly chosen for training, an even more chaotic effect is observed in the non-IID case. Lastly, sampling fractions of the clients for training, instead of all of them, and implementing

centralized evaluation was concluded to be ways in which communication efficiency could be improved.

VIII. FUTURE WORK

Even though using FL handles many of the privacy concerns of traditional learning methods, there are still means by which some training data can be accessed from the models sent to the server through model inversion. This was shown in [22], where it was described how a model inversion attack was able to recover training data from facial recognition systems. Hence, future work could investigate differential privacy training of the neural networks. The concept, as written in [23], counteracts possibilities of model inversion and increases the protection of the models, allowing them to be created on-device and sent to a server while maintaining privacy.

Furthermore, there has been a recent development in response to the shortcomings that were observed when using FL on heterogeneous data, mentioned in subsection IV-A2. A new FL framework by the name *HeteroFL*, first presented in [24], was used to train heterogeneous local models with different computational characteristics while still outputting a single global inference model. On the same topic of heterogeneous data, a method named *FedDyn* can also be used to make losses recorded on client devices converge to the global loss. This method was first mentioned in [25] and describes that by handling the heterogeneity of the data, full minimization of loss can be achieved on each device.

Consequently, exploring one or multiple of the concepts above in future works concerning this bachelor's degree thesis is of interest to further develop and improve FL and ultimately make it more well-rounded and applicable for different uses and scenarios in society.

APPENDIX GITHUB REPOSITORY

This is a link to the GitHub repository containing all necessary docker files and programs for simulations: <https://github.com/Zigolox/Federated-Learning-On-RPi>

ACKNOWLEDGMENT

We would like to thank our supervisor, Dr. Hao Chen, for showing interest and enthusiasm for our work and for guiding us in the research process.

REFERENCES

- [1] H. S. Lallie *et al.*, "Cyber security in the age of covid-19: A timeline and analysis of cyber-crime and cyber-attacks during the pandemic," *Computers & Security*, vol. 105, p. 102248, Mar 2021.
- [2] K. Fowler, *Data Breach Preparation and Response: Breaches are Certain, Impact is Not*, 1st ed. CA, Toronto: Syngress, 2016.
- [3] (2019, Apr) Data on 540 million facebook users exposed. [Online]. Available: <https://www.bbc.com/news/technology-47812470>
- [4] J. Konečný *et al.*, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, Oct 2016.
- [5] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, Dec 2015.
- [6] Aphex34. (2016, Dec) Typical cnn architecture. [Online]. Available: https://commons.wikimedia.org/wiki/File:Typical_cnn.png
- [7] Jeromemetronome. (2019, Jun) Federated learning process in central orchestrator case. [Online]. Available: https://commons.wikimedia.org/wiki/File:Federated_learning_process_central_case.png

- [8] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning," *Advances in neural information processing systems*, vol. 20, Dec 2007.
- [9] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, Feb 2017, pp. 1273–1282.
- [10] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, Dec 2019.
- [11] D. J. Beutel *et al.*, "Flower: A friendly federated learning research framework," *arXiv preprint arXiv:2007.14390*, Mar 2020.
- [12] 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, Nov 1998.
- [13] J. Steppan. (2017, Dec) A few samples from the mnist test dataset. [Online]. Available: <https://commons.wikimedia.org/wiki/File:MnistExamples.png>
- [14] T. Darrell, M. Kloft, M. Pontil, G. Rätsch, and E. Rodner, "Machine learning with interdependent and non-identically distributed data," in *Dagstuhl Reports*, vol. 5, no. 4. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Apr 2015.
- [15] (2022, apr.) Package software into standardized units for development, shipment and deployment. [Online]. Available: <https://www.docker.com/resources/what-container/>
- [16] (2022, apr.) What is a raspberry pi? [Online]. Available: <https://opensource.com/resources/raspberry-pi>
- [17] (2022, apr.) Raspberry pi 4 tech specs. [Online]. Available: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/>
- [18] M. H. ("Laserlicht"). (2019, Jul) Raspberry pi 4 model b from the side. [Online]. Available: https://commons.wikimedia.org/wiki/File:Raspberry_Pi_4_Model_B_-_Side.jpg
- [19] (2022, apr.) Linear. [Online]. Available: <https://pytorch.org/docs/stable/generated/torch.nn.Linear.html>
- [20] (2022, apr.) Dropout. [Online]. Available: <https://pytorch.org/docs/stable/generated/torch.nn.Dropout.html>
- [21] (2022, apr.) Evaluation. [Online]. Available: <https://flower.dev/docs/evaluation.html?highlight=evaluation>
- [22] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1322–1333.
- [23] M. Abadi *et al.*, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, Oct 2016, pp. 308–318.
- [24] E. Diao, J. Ding, and V. Tarokh, "Heterofl: Computation and communication efficient federated learning for heterogeneous clients," *arXiv preprint arXiv:2010.01264*, Dec 2020.
- [25] D. A. E. Acar *et al.*, "Federated learning based on dynamic regularization," *arXiv preprint arXiv:2111.04263*, Nov 2021.

CONTEXT N – PART II

INFORMATION ENGINEERING: BIG DATA & AI

POPULAR DESCRIPTION

Artificially intelligent or just plain stupid?

When thinking about Artificial Intelligence (AI) either imagery of tyrannical supercomputers devoid of emotion conquering humanity, or sleek robots catering to our every whim, springs to mind. The reality is more likely a third option; simple machines that are extraordinary at very specific tasks, but subpar at everything that falls outside what their limited 'brain' can comprehend.

There seems to be a commonly held belief that AIs will soon exhibit and surpass human-like abilities. This belief stems, not from provable facts, but from conflating artificial and human intelligence. But using the one, to judge the other, might not be the best approach.

Consider the classification of bird species. It is easy to program a computer to know all known species, whereas the average human knows comparably few. If you train an AI, using pictures of birds, machine beats human every time. Most humans would think it's a *difficult* task. If you now show the network and the human a picture of a dragon, the network will still try to classify it as a bird, while the human *easily* classifies it as "not a bird". Current AI models can not conceive of concepts beyond what they've been taught.

Much like how an idiot can still be useful we can still make great use of AI technology. Need to find the fastest route to work? AI can help. Need to recognize handwritten text? AI to the rescue! Doctors can't be bothered to look through hundreds of x-ray images? AI will be happy to do it for them. Narrowing down the functionality of the AI might kill the dream of robot butlers, but it will allow them to perform specific tasks excellently.

SUMMARY OF PROJECT RESULTS

A hot button topic within AI research is the development of machine learning algorithms. There is a perceived goal within industry and research concerns to apply machine learning algorithms to solve problems which so far only humans could tackle due to the complexity of the questions involved. One of the main areas where machine learning is starting to be utilized is within medicine, both to diagnose patients and for further research. The biggest limiting factor in machine learning is the availability of large amounts of varied data to train on. Since no dataset can cover all possible scenarios that can be encountered in practice, it is also of interest how the trained models perform when introduced to data that differ from the training set. 'There ain't no such thing as a free lunch', especially in machine learning; a machine learning algorithm will only perform well on tasks and data it is designed for.

Group N5 examined a discriminative and generative model for image classification. A convolutional neural network was used as the discriminative classifier and a flow-based model using RealNVP was used as the generative classifier. The models were trained and evaluated on the MNIST dataset which consists of handwritten digits from 0 to 9. Project group N5 compared the accuracy of the two image classifiers on images with and without added noise. The effect on accuracy of the models was also evaluated with models trained on noisy images.

In project N6, the concept of Big Data was applied within the medical field. In a collaboration between Karolinska Institutet and KTH, data has been collected from the neonatal intensive care unit. Many algorithms are in development to improve the time to diagnose and the accuracy of diagnoses. In this project, we aimed to study how well random forest algorithms could

detect sepsis, a possibly life-threatening condition characterized by a dysregulated immune response to blood infection. If the results are satisfactory, the aim is to eventually apply the algorithms to aid setting diagnoses.

The goal of project N7 was to test the robustness of modern convolutional neural networks. The aim was to see how a current image classification model performed when further trained on a different dataset, namely when the testing images differ slightly from the training images. The dataset used for further training was a relatively small set of images depicting fruits in different visibility conditions. The SqueezeNet model was further trained on a part of the new dataset consisting of images with the same level of obscurity and was then tested on images where the obscurity was less and more intense. The results show that the fully trained model is less accurate on both images with less obscurity and images with more obscurity. This led to the conclusion that modern convolutional neural networks perform worse on images that differ from the training images, even if the difference is that the testing images are much less obscured than the training data and would therefore be classified as 'easier' by a human spectator.

In project N8 the aim was to develop a machine learning model to improve a recent DNA sequencing method called *nanopore sequencing*. This method involves measuring the current through a DNA molecule as it passes through a nanoscopic hole. The different nucleobases of the DNA will lead to different measured current signals due to their conductive properties. From the recorded signal, a genetic sequence is derived. Though still in a development stage, nanopore sequencing could prove to be a more efficient and cheaper alternative to conventional sequencing methods, as the overall physical process of recording the DNA data is simpler and less time consuming. The models built in this project were designed to predict the accuracy of a sequence derived from a given measurement. This could then be used to determine which measurements of a molecule to use when producing a new sequence. Different types of models, including linear regression and neural networks were tested and compared with regard to their relative accuracy and overall effectiveness.

Machine learning systems used today are designed for limited tasks and fall in the category of narrow AI. Huge amounts of man- and computing power are spent improving the technology, and not without impressive results: algorithms are outperforming humans in multiple areas. The applications of narrow AI are broad. High level systems are integral to a lot of cutting edge technology, such as: computer vision and AI assisted medical diagnostics. In some areas computers excel, in large part due to the increased availability of larger datasets.

With better access to more varied data to train on, a topic for future studies could be to train algorithms using larger and more varied datasets. Otherwise, new models need to be developed that are better at generalizing what they learn to tackle similar problems with slightly different data. If this can not be done perhaps research should instead be focused on narrow AI, where the models are expected to do one thing, and do that one thing perfectly.

IMPACT ON SOCIETY AND ENVIRONMENT

Artificial intelligence has in later years prominently made its way into multiple sectors of society, and has in many regards revolutionized the way we make predictions and draw conclusions. The rapidly accelerating integration of artificial intelligence has however also become cause for concern. The problems with AI exist at many different levels, from the general population being expected to trust decisions made by a fundamentally imperspicuous system, to AI developers having to take great care to create well-defined systems that take into account all the nuances of the real world. The use of AI to predict and make use of sensitive data highlights already relevant societal issues regarding privacy and personal security. Expecting AI to make morally weighted decisions also calls into question the way we look at accountability and ethics. There would need to be a system of responsibility in case an AI makes a decision that is generally seen as morally incorrect.

The implementation of Big Data and AI will lead to huge changes in our society. With the increased amount of data available, as well as the increased utilization of AI systems, there are also more AI programs which are able to make decisions faster and more efficiently than humans are able to. Similarly to how transistors reduced the cost of computing, and how the advent of the internet reduced cost of information, the advent of Big Data and AI will lead to reduced cost of prediction. For example, autonomous cars will lead to safer driving and lower emissions due to more efficient driving.

AI systems may reduce the cost of prediction, but could increase the cost for the environment as well. Storing huge amounts

of data and training large scale AI systems consumes a lot of energy. This is in great part due to energy inefficient training and storage methods. To counteract this more energy efficient training and storage methods should be developed in the future.

Another issue with the use of AI is surveillance. Around the world there is a huge surveillance infrastructure with cameras able to map, track and consequently control people by enforcing certain policies. The addition of real time analysis by AI systems seeks to exponentially expand the reach of these systems. On one hand surveillance could be used for positive things like detecting a school shooter. On the other hand, some autocratic countries like Saudi Arabia, China and Russia are using this technology for mass surveillance. We believe this to be a violation of human rights and see implementation of AI technology in surveillance as largely negative.

One of the limitations with AI models is the type of data and methods used in training. Biases of the developers may unintentionally be reflected within the models. The results that are produced can also be indirectly controlled by cherry picking data. A model only trained on dogs will likely have trouble identifying other animals. Problems occur if biased models are used to categorize groups of people; job applicants or people who are likely to develop certain diseases, to name a few examples. The process for training models must be transparent and describe what factors are used for different decisions. Thought must also be put into who is responsible for a biased model.

It is difficult to determine the individual who should be accountable for the action which an AI takes. When an AI makes a decision, who is responsible for that action? Is it the engineer who developed the AI, the user, governmental lack of regulations, or is it someone else? Determining this prior to implementation is essential. With the utilization of AIs in warfare, who is held accountable for lives taken by the AI? When autonomous vehicles are implemented, who is responsible when an accident happens? There are no clear answers for these questions and we believe that, because of this, governments and private companies alike, have a responsibility to make sure that such AI does not reach the market.

AI in medicine can massively assist physicians in making their diagnosis and choosing their method of treatment. Additionally, more resources can be allocated to treatment instead of the diagnosis. However, the question arises who is responsible in case of a false AI diagnosis that leads to permanent damage or death of a person. We need to ask ourselves whether or not we should treat the algorithm the same as a person. If an AI and a physician have differing opinions on a diagnosis, a decision needs to be made regarding who to listen to, and whether the patient should be given the option to choose whether they want an AI's opinion or a physician's.

Many of the project groups in this context have focused on the explainability of the algorithm, and for good reason. A lack of trust in AI systems by for example doctors can result in an underuse of life saving tools. Automated decision making obfuscates the chain of accountability and pre-existing bias may be replicated. Explainability is a tool to try to mitigate these problems.

Overall, we think that AI will have a great impact on society because it will reduce the cost of prediction. However, some of the aspects of AI are extremely dangerous and we have to be careful so that governments do not use the technology to enforce inhumane policies on the population.

Comparison of Discriminative and Generative Image Classifiers

Simon Budh and William Grip

Abstract—In this report a discriminative and a generative image classifier, used for classification of images with handwritten digits from zero to nine, are compared. The aim of this project was to compare the accuracy of the two classifiers in absence and presence of perturbations to the images. This report describes the architectures and training of the classifiers using PyTorch. Images were perturbed in four ways for the comparison. The first perturbation was a model-specific attack that perturbed images to maximize likelihood of misclassification. The other three image perturbations changed pixels in a stochastic fashion. Furthermore, The influence of training using perturbed images on the robustness of the classifier, against image perturbations, was studied. The conclusions drawn in this report was that the accuracy of the two classifiers on unperturbed images was similar and the generative classifier was more robust against the model-specific attack. Also, the discriminative classifier was more robust against the stochastic noise and was significantly more robust against image perturbations when trained on perturbed images.

Sammanfattning—I den här rapporten jämförs en diskriminativ och en generativ bildklassificerare, som används för klassificering av bilder med handskrivna siffror från noll till nio. Syftet med detta projekt var att jämföra träffsäkerheten hos de två klassificerarna med och utan störningar i bilderna. Denna rapport beskriver arkitekturerna och träningen av klassificerarna med hjälp av PyTorch. Bilder förvrängdes på fyra sätt för jämförelsen. Den första bildförvrängningen var en modellspecifik attack som förvrängde bilder för att maximera sannolikheten för felklassificering. De andra tre bildförvrängningarna ändrade pixlar på ett stokastiskt sätt. Dessutom studerades inverkan av träning med störda bilder på klassificerarens robusthet mot bildstörningar. Slutsatserna som drogs i denna rapport är att träffsäkerheten hos de två klassificerarna på oförvrängda bilder var likartad och att den generativa klassificeraren var mer robust mot den modellspecifika attacken. Dessutom var den diskriminativa klassificeraren mer robust mot slumpmässiga bildförvrängningar och var betydligt mer robust mot bildstörningar när den tränades på förvrängda bilder.

Index Terms—Image classification, CNN, Normalizing flows, RealNVP, Adversarial examples

Supervisors: Anubhab Ghosh and Saikat Chatterjee

TRITA number: TRITA-EECS-EX-2022:173

I. INTRODUCTION

In the past century many disruptive technologies have emerged. The invention of the transistor reduced the cost of computing significantly and the creation of the internet reduced the cost of information considerably. In the next century machine learning (ML) could possibly be a disruptive technology that significantly reduces the cost of prediction.

A fundamental task within machine learning is image classification. A machine learning algorithm used for image

classification should be able to provide a (correct) prediction of the image content for a given image, described in [1, p. 98]. A discriminative model utilizing a convolutional neural network (CNN) [1, pp. 326-366] is often used for classification tasks. With increasing data sets and increasing computational power, new CNN architectures are continuously developed that push the state of the art in image classification, shown in [2]. However, images to be classified can contain perturbations, which usually decrease the accuracy of the classifier demonstrated in [3] and [4]. Another approach to image classification is to use a maximum likelihood classifier based on generative models such as normalizing flows [5]. As suggested in [6], the generative based classifier could potentially be more robust against image perturbations than a discriminative classifier.

The purpose of this paper is to compare the performance of a discriminative and a generative image classifier. The classifiers classify images from the MNIST database [7], which contain handwritten digits from zero to nine. For the comparison, the two classifiers have at least 90% accuracy on unperturbed images not used in training. The discriminative classifier is implemented using a convolutional neural network and the generative classifier is implemented using normalizing flows. The classifiers accuracy on images with and without image perturbations are compared. Before implementing image classifiers, a discriminative and a generative classifier are implemented to classify toy datasets two moons and two circles from scikit-learn [8]. This is to verify the normalizing flows-based classifier and to compare the classification performance.

II. THEORY

A. Discriminative and Generative Image Classifiers

Discriminative classifiers learn distinguishing features that separate different classes of images. Images that exhibit features of a certain class should have a high likelihood to be classified as that class. The discriminative classifier draws boundaries between the different classes, which are known as decision boundaries. The discriminative classifier is trained using a supervised learning setup, where the classes of the training images are known. This enables the classifier to learn distinctions between the different classes, explained in [1, pp. 96-98]. Generalization is thereafter the classifier's ability to classify images that have not been seen in training.

Generative models [9] also learn features of the different image classes. However, the generative models try to learn the underlying distribution of the different classes. A maximum likelihood classifier based on generative models is trained using an unsupervised learning setup and consists of one generative model for each image class. In training, each

generative model receives images of one class and learns the characteristics of that class. The maximum likelihood classifier based on generative models is referred to as generative classifier in this paper. When classifying an image using maximum likelihood classification [10], the image is classified as the class with the highest probability. The image will be passed to each generative model which returns the probability that the image was produced by the class. Then the image is labeled as the class of which corresponding model returns the highest probability. In other words the image is classified as the class of which is most likely to produce the image. Generative models are also capable of generating images based on the characteristics learned in training.

B. Deep Feedforward Neural Networks

The discriminative and generative classifiers used in this report are implemented with deep feedforward neural networks (DFNN). Machine learning models used for classification are designed to map inputs \mathbf{x} to a list of classes \mathbf{y} . In mathematical terms: $f^*(\mathbf{x}) \mapsto \mathbf{y}$ where f^* is learned in training of the model described in [1, p. 164].

The term "neural" in DFNN refers to the structure of the network, which is inspired by brain neurons which are represented as nodes in the network. The nodes are organized in layers, meaning a layer is a group of nodes. In a DFNN data flows from the input through weighted connections between the layers to the output. The term "deep" in DFNN refers to the fact that the network consists of several layers. The input and output layers are considered visible. Between input and output are the hidden layers, where the model transforms the input data, explained in [1, pp. 164-167].

C. Model Training

In training, weights in the network are adjusted to minimize the loss. Loss [1, pp. 271-272] measures the model's performance and is reduced as the model gets more accurate. The weights are adjusted using an optimizer [1, pp. 271-273] with a learning rate, which determines how vast the adjustments should be. A pass through the training dataset is called an epoch. Sometimes, the learning rate is decreased over training epochs so weights are adjusted vastly at first and slightly towards the end. This is to ensure that learning from earlier training does not get lost and can be done with weight decay. The amounts of epochs in training are chosen to neither overfit nor underfit. Overfit is when the model is not able to generalize to unseen data and underfit is when the model is not able to learn from training data, described in [1, pp. 224-225]. A dataset is usually split into training data and validation data, where training data is 80% of the dataset and validation data is 20% of the data set, described in [1, p. 119]. This is to validate generalization of the model, by testing the model on data not used in training.

D. Activation Functions

Activation functions [11] are applied between each layer in the networks to introduce nonlinearity. DFNN needs to learn complex mappings from input to output, which often are nonlinear. Some examples of activation functions are rectified linear unit [12], tanh, and sigmoid [13].

E. Autoencoder

An autoencoder compresses higher dimensional input data to a lower dimensional embedding, without losing important features described in [1, pp. 499-501]. The compressing of data is called encoding and the opposite, mapping embedding to input data, decoding. To reduce computing costs images are often encoded to embeddings. The autoencoder is implemented using a neural network and learns to identify important features of input data in training.

F. Convolutional Neural Networks

Convolutional neural networks (CNNs) are commonly used in image classifiers. These are networks that contain convolutional layers. The layers contain a kernel [1, p. 326], which in the two-dimensional case is a matrix with m rows and n columns. In the convolutional layers, the output S is calculated using cross-correlation between the kernel K and input I :

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i+m, j+n) K(m, n) \quad (1)$$

described in [1, p. 329]. The convolutional layers work as filters that extract distinguishing features of images.

CNNs can employ pooling [1, pp. 335-339] operations to make the model invariant to small input changes, such as positional changes of features. This is useful for image classification as features position can differ between images. One example of pooling is max pooling [1, p. 335].

Stride can be used as an alternative to pooling, presented in [14]. The stride of a convolutional layer determines the starting position of the next cross-correlation.

Padding adds transparent pixels around the borders of an image. Without padding images will shrink at each convolutional layer, which will lead to information loss, explained in [15].

G. Normalizing Flows

Normalizing flows seeks to map simple probability distributions to complex probability distributions. The change of variables formula can be used to evaluate densities of a stochastic variable, of which is a deterministic transformation of another stochastic variable. Let X and Z be stochastic variables which are related by the invertible transformation $f : \mathbb{R}^n \mapsto \mathbb{R}^n$ in a way that $X = f(Z)$ and $Z = f^{-1}(X)$. Then the change of variables formula is given by:

$$p_X(x) = p_Z(f^{-1}(x)) \left| \det \left(\frac{\partial f^{-1}(x)}{\partial x} \right) \right| \quad (2)$$

as formulated in [16], the determinant is of the Jacobian matrix.

The "flow" in normalizing flows is the invertible transformation. The transformation can consist of many invertible transformations to generate a very complex invertible transformation. There are some requirements on the normalizing flows model; input and output must have the same structure, the transformation should be invertible and computation of the Jacobian determinant must be efficient, stated in [16].

H. RealNVP

Real Non-Volume Preserving (RealNVP) is a normalizing flows model used in this project. The model consists of two invertible transformations, rescaling layers and additive coupling layers. It is possible to build a bijective function by combining many simple bijective functions as shown in [16]. A simple bijection can be referred to as a coupling layer. For a N -dimensional input \mathbf{x} and $n < N$ (where $n \geq \frac{N}{2}$ and therefore $N \geq 2$), the output \mathbf{y} from a coupling layer is given by:

$$\begin{aligned} y_{1:n} &= x_{1:n} \\ y_{n+1:N} &= x_{n+1:N} \otimes \exp(s(x_{1:n}) + t((x_{1:n}))) \end{aligned} \quad (3)$$

where \otimes is the element-wise product, t and s is for translation and scale, as described in [16]. The $s(\cdot)$ and $t(\cdot)$ functions $\mathbb{R}^n \mapsto \mathbb{R}^{N-n}$ are implemented using neural networks. The coupling layers do not change all components in the input. Therefore coupling layers are usually connected in an alternating pattern where the unaffected components are changed in the next layer, as described in [16]. The Jacobian for this transformation is:

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}^T} = \begin{bmatrix} I_n & 0 \\ \frac{\partial y_{n+1:N}}{\partial x_{1:n}^T} & \text{diag}(\exp[s(x_{1:n})]) \end{bmatrix} \quad (4)$$

where I_n is the identity matrix of size n and $\text{diag}(\exp[s(x_{1:n})])$ is a diagonal matrix with diagonal $\exp[s(x_{1:n})]$, proved in [16]. Because the Jacobian is triangular, the determinant can be computed efficiently and is $\exp[\sum_j s(x_{1:n})_j]$. However, the Jacobian of $s(\cdot)$ and $t(\cdot)$ is not needed and they can therefore be neural networks.

The partitioning can be achieved with a binary mask \mathbf{b} , \mathbf{y} is then given by:

$$\mathbf{y} = \mathbf{b} \otimes \mathbf{x} + (1 - \mathbf{b}) \otimes (\mathbf{x} \otimes \exp(s(\mathbf{b} \otimes \mathbf{x})) + t(\mathbf{b} \otimes \mathbf{x})) \quad (5)$$

as shown in [16]. To avoid instability in training the output from coupling layers are normalized. The rescaling function of \mathbf{x} (maps to $\hat{\mathbf{x}}$), using estimated batch mean $\tilde{\mu}$ and variance $\tilde{\sigma}^2$, is given by:

$$\hat{\mathbf{x}} = \frac{\mathbf{x} - \tilde{\mu}}{\sqrt{\tilde{\sigma}^2 + \psi}} \quad (6)$$

where ψ is an arbitrary small constant and the Jacobian determinant is $(\prod_i (\tilde{\sigma}_i^2 + \epsilon))^{-\frac{1}{2}}$, declared in [16]. The batch normalization behaves like a linear rescaling on every dimension.

I. Adversarial Examples

Perturbing an image imperceptibly can affect the performance of an image classifier significantly, shown in [3]. The purpose of an adversarial attack is to perturb the image imperceptibly to cause a misclassification, described in [3].

J. Fast Gradient Sign Attack

Fast gradient sign attack (FGSM) is an attack that seeks to achieve misclassification, described in [3]. FGSM attacks a model through the learning process by using the model's gradients. In training the model will try to minimize loss

by adjusting weights based on backpropagation of gradients. In the attack, instead of adjusting weights, the attacker will perturb the input image to maximize loss based on the backpropagation, described in [3].

III. METHOD

The discriminative and generative classifiers used in this project had some architecture and training parameters chosen. In this project the main focus was to compare two types of classifiers. The parameter choices were not the purpose nor a priority in this project, however all parameters can be found under section IV. For the comparison the classifiers should have at least 90% accuracy on unseen data free from perturbations, which all classifiers achieved and therefore the parameters were sufficient for the purpose of this project.

A. Discriminative Toy Data Classifier

Two discriminative classifiers were trained, one for two moons and one for two circles. The structure of the discriminative classifiers was a fully connected network with two hidden layers. The activation function was rectified linear unit [12], the loss function used was cross entropy loss [17] and the optimizer was stochastic gradient descent [18]. A softmax function was used on the two output channels, which rescaled the outputs to the range $[0, 1]$ and the sum of the outputs was 1. In other words the softmax function converted numerical outputs to probabilities. For each epoch in training data points were taken from the toy dataset, of which 20% was used for validation and the remaining 80% was used for training. The train data points were passed to the model and weights were adjusted based on the loss. Then the model was validated on the validation data, to ensure the model generalized to unseen data. In validation the weights were not updated, but the validation loss was saved. Before training the classifiers, a long test run with many epochs was performed. After a certain amount of epochs the validation loss stopped decreasing in the test run. The same amount of epochs was then used to train the model, to neither overfit nor underfit. Then the models were tested on unseen test data and the decision boundary was determined. Then the test data was perturbed with Gaussian noise with increasing standard deviation and accuracy was measured. In Fig. 1 the two toy datasets and how the toy datasets are affected by Gaussian noise are visualized.

B. Generative Toy Data Classifier

Two generative classifiers were trained, one for two moons and one for two circles. The architecture of the generative classifiers was inspired by architectures used in [19]. The generative classifiers $s(\cdot)$ and $t(\cdot)$ were composed of fully connected deep neural networks. The networks had two input and output channels as well as two hidden layers. The activation function for the neural networks was leaky rectified linear unit [20]. The output layer of $s(\cdot)$ had tanh as activation function. The optimizer used was Adam [21] and the loss function was the negative log probability.

Training of the generative classifier was a bit different compared to the discriminative classifier, the generative model

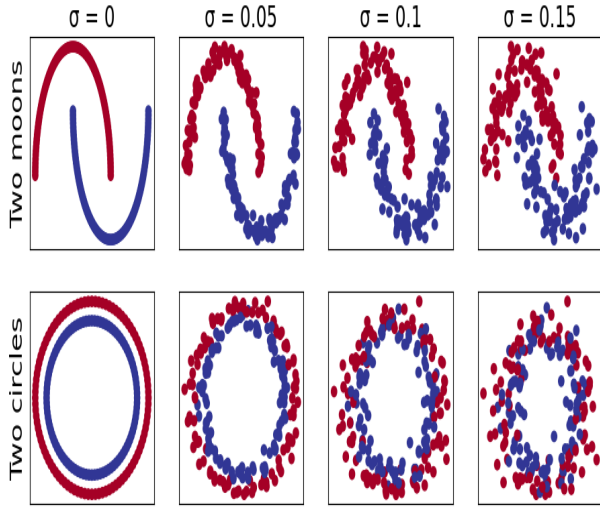


Fig. 1. Visualization of toy datasets perturbed with Gaussian noise for different standard deviation σ of the Gaussian noise

consisted of two models. One model learned the structure of one moon or circle and the other model learned the structure of the other moon or circle. To predict a data point the point was given to both models, which returned a probability that the point was generated by the model. The point will then be classified using maximum likelihood, the point was classified as the class of which model gave the highest probability. In training the data points of the two shapes are separated and given to the two models. The amount of training epochs was chosen with respect to minimizing training loss. When training loss no longer decreased, the training was terminated to neither overfit nor underfit. The generative classifiers were then tested on unseen data. Then test data was perturbed with Gaussian noise and accuracy was measured for increasing standard deviation of the noise. In Fig. 1 the two toy datasets and how the toy datasets are affected by Gaussian noise are visualized.

C. Discriminative Image Classifier

The architecture of the discriminative image classifier was inspired by the architecture used in [22]. The discriminative classifier was implemented using a CNN. The architecture for the CNN consisted of one input channel, two hidden convolutional layers and ten output channels. After the first convolutional layer the activation function rectified linear unit [12] was used and then max pooling was applied. The second convolutional layer was connected to a fully connected layer with the 10 output channels. The 10 output values were values for the 10 different classes. When predicting an image label, the label was the corresponding output channel with the largest value.

The optimizer used in training was stochastic gradient descent [18] and the loss function was cross entropy loss [17]. The classifier was trained for a large number of epochs and one classifier was saved after each epoch. Then the classifier with the smallest training loss was chosen for the comparison.

D. Generative Image Classifier

The architecture of the generative classifier was inspired by the architecture used in [23]. The generative model was a variational autoencoder, which used an autoencoder to convert input images to an embedding. The images had size 28x28 and the embedding had size 1x20. The RealNVP model learned the mapping from embedding to a simple distribution, instead of mapping directly from an image to a simple distribution. To learn a mapping directly from the images to a simple distribution would require a more complex architecture and would be materially more computationally expensive.

The encoding of the autoencoder was implemented with a CNN, which had two convolutional layers connected to each other. The second convolutional layer was connected to a fully connected layer, of which output was the embedding. The activation function used in the CNN of the autoencoder was rectified linear unit [12]. The decoding of the autoencoder was implemented with the same architecture as the encoding, but in reverse. In training, each image was passed through the encoder layers to create an embedding and then the embedding was passed through the decoding layers. The sigmoid function [13] was applied to the output of the decoding layers to ensure the reconstructed image pixels were in the range $[0, 1]$. The loss was calculated with respect to the reconstruction error with binary cross entropy loss [24]. The optimizer used was Adam [21].

The normalizing flows model using RealNVP was constructed with nine coupling layers. The coupling layers utilized rectified linear unit [12] as activation function. The generative classifier needed 10 different models, one model for each class of handwritten digits. For each model, one class of images was separated and encoded. Then the model learnt the mapping between embedding and a simple probability distribution in training. The optimizer used was Adam [21] and the loss function was the negative log probability. After an amount of epochs the training loss stopped decreasing, at which point the training should be aborted to neither underfit nor overfit.

After the training of all models a maximum likelihood classifier was created. The classifier passed an image through all models, then the image was classified as the class of which corresponding model gave the highest probability.

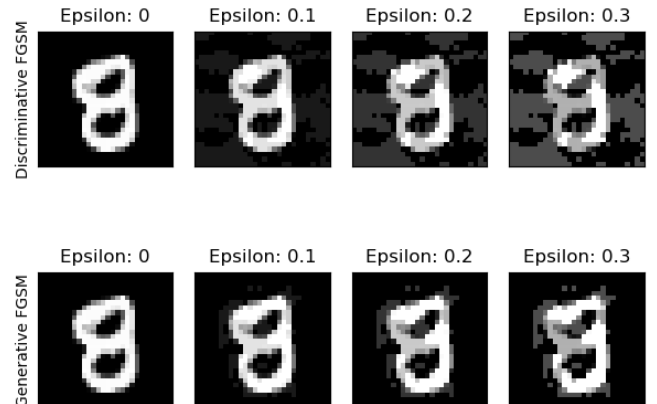


Fig. 2. Visualization of image perturbations caused by the FGSM attack on the discriminative and generative classifiers for different weights ϵ of perturbation applied to the test images

E. FGSM Attack on the Image Classifiers

After the two image classifiers were trained, the classifiers were attacked with a FGSM attack. The FGSM attack was performed by passing an image to the classifier and then computing the log probabilities for the different classes. Then the negative log likelihood loss [25] was passed back to the models. Instead of updating the weights in the models, the gradients were used to perturb the image in the worst possible way to increase the loss. The perturbation was a matrix of 28×28 with values -1 and 1. -1 made pixels darker and 1 made pixels brighter in the perturbed image. This perturbation matrix was then applied to the image with a weight ϵ . The resulting perturbation of the images is shown in Fig. 2.

F. Evaluation on Gaussian and Impulse Image Perturbations

The two classifiers were evaluated on stochastic image perturbations. First, images were perturbed with additive white Gaussian noise and accuracy was measured for different variances of the Gaussian noise. Secondly, images were perturbed by negative impulse noise [26]. This was done by randomly selecting some pixels and adding -1 to them, which results in the pixels being set to black. Accuracy for the classifiers was measured for different percentages of the total pixels perturbed by negative impulse noise. The third stochastic noise was positive impulse noise [26]. This was similar to negative impulse noise, but instead 1 was added to the pixels, which set the pixels to white. The accuracy of the classifiers was measured for different percentages of the total pixels affected by positive impulse noise. Visualization of the three stochastic image perturbations are shown in Fig. 3.

G. Training Classifiers With Image Perturbations

In the previous sections the image classifiers were trained on images free from perturbations. In this section the influence of perturbing some of the training images was studied. A new set of discriminative and generative classifiers for each of the stochastic noises were trained. The classifiers had the same

architecture and were trained the same way as the classifiers trained on images free from perturbations. In training, half of the training images were perturbed with stochastic noise and images were passed to the models in a randomized order. Then the classifiers were compared to the corresponding classifier, trained without image perturbations, on images containing the same type of perturbations used in training.

IV. EXPERIMENTAL SETUP

PyTorch [27] machine learning frameworks were used to implement and train the models in this project. The datasets used in this project were two toy datasets, two moons and two circles, from scikit-learn [8] as well as the MNIST database [7]. The toy datasets contained data points in the two-dimensional Cartesian coordinate system. The toy data formed a shape, either a circle or a moon, for each class. The MNIST dataset contained 70000 images of handwritten digits from zero to nine. The images were matrices, containing 28 rows and 28 columns, with entries for each pixel. The pixels had values between 0 and 1, where 0 was black and 1 was white.

A. Discriminative Toy Data Classifier

The fully connected network had two hidden layers, with 15 nodes each, between the two input and two output channels. The chosen learning rate for the optimizer was 0.1. In each epoch during training 1000 data points were taken from the toy dataset, of which 200 was separated for validation and the remaining 800 points were used for training. The test run before training the classifiers was 1000 epochs long. From the test run it was decided that the discriminative classifier should be trained for 800 epochs on two moons and 720 epochs on two circles. The testing used 1000 unseen data points free from perturbations.

B. Generative Toy Data Classifier

The neural networks that implemented the $s(\cdot)$ and $t(\cdot)$ functions had two hidden layers with 256 nodes each between the two input and two output channels. The mask used had a checkerboard pattern, where half the positions had value 0 and other half had value 1, meaning n was chosen as $n = \frac{N}{2}$. The learning rate of the optimizer was 10^{-4} and the learning rate decayed 11% every epoch. In each epoch of training, 50000 data points were given to each generative model. The generative classifier was trained for 240 epochs on two moons and 242 epochs on two circles, which was decided using a test run as for the discriminative classifier. The testing used 1000 unseen data points free from perturbations.

C. Discriminative Image Classifier

The first convolutional layer in the CNN had 16 output channels, kernel size five, convolution stride one and padding was two. The max pooling after the first convolutional layer had kernel size two. The second convolution layer had 16 input channels and 32 output channels. The kernel size, convolution stride, padding, activation function and pooling was the same for the second convolutional layer as for the first convolutional layer. In every train epoch, 60000 images were given to the classifier. The learning rate of the optimizer was 0.01 and

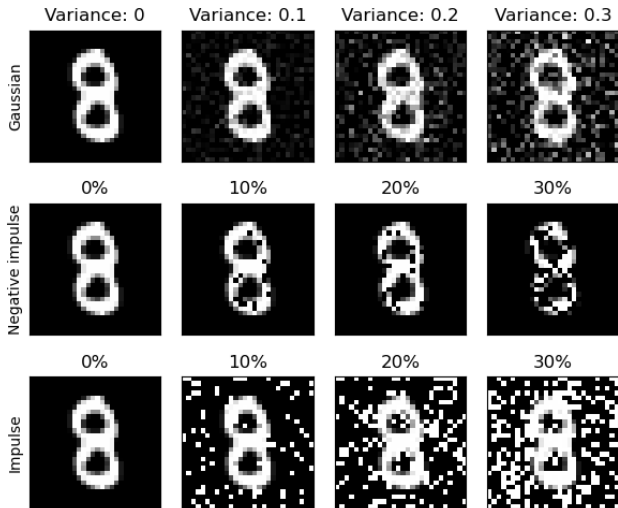


Fig. 3. Visualization of images perturbed with stochastic noise

decayed 11% each epoch. The classifier was trained for 50 epochs and the classifier with the smallest training loss was selected for the comparison. The test on unseen images free from perturbations had 10000 images.

D. Generative Image Classifier

The first convolutional layer of the autoencoders' CNN had one input channel and 32 output channels. The stride for the convolutional layer was two, the size of the convolving kernel was three and the padding was one. The 32 output channels were connected to the second convolutional layer, which had 64 output channels. The second convolutional layer had a stride of two, kernel size three and padding of one. The output channels of the second convolutional layer were connected to a fully connected layer, of which output was the embedding. The autoencoder was trained for 10 epochs and in each epoch the autoencoder was given 60000 images from the MNIST dataset. The learning rate of the optimizer was 10^{-3} and weight decayed 10^{-5} each epoch.

Each coupling layer of the RealNVP model had one hidden fully connected layer with 200 nodes. The mask used by the generative model had alternating values 0 and 1 starting with 0 at the first position of the embedding, in other words n was chosen as $n = \frac{N}{2}$. Each class of images had approximately 6000 images, which was used every epoch in training. The learning rate of the optimizer was 10^{-4} and weight decay each epoch was 10^{-5} . To determine the amount of epochs in training a test run of 200 epochs was done. From the test run, it was determined to train the generative models for 30 epochs. The test on unseen images free from perturbations had 10000 images.

E. Training Image Classifiers With Image Perturbations

The Gaussian noise added to half of the train images had variance 0.5. The negative impulse noise added during training was added to 70% of pixels, randomly chosen, in the train images. The impulse noise added in training was added to 10% of pixels, randomly selected, in the train images.

V. RESULTS

A. Accuracy of Toy Data Classifiers

TABLE I
TOY DATA CLASSIFIERS' ACCURACY ON TEST DATA

Dataset	Classifier	Accuracy
Two moons	Discriminative	100%
Two moons	Generative	98.9%
Two circles	Discriminative	100%
Two circles	Generative	92.8%

The toy data classifiers' accuracy on unseen data points for the two toy datasets are shown in Table I. The decision boundary determined by the discriminative classifier for the toy datasets are illustrated in Fig. 4 for two moons and Fig. 6 for two circles. The modeling of the toy data classes location by the generative classifier are shown in Fig. 5 for two moons and Fig. 7 for two circles.

In Fig. 8 the performance of the classifiers when toy data have been perturbed with Gaussian noise, for increasing

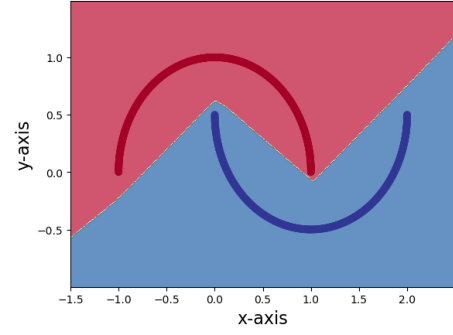


Fig. 4. Decision boundary drawn by the two moons discriminative classifier on test data points without perturbations

variance, is shown. The discriminative classifier was a bit more robust against Gaussian noise perturbations on two moons than the generative classifier. On two circles, there was no significant difference in accuracy on perturbed data between the classifiers.

B. Accuracy of Image Classifiers on Test Images Without Perturbations

TABLE II
CLASSIFIERS' ACCURACY ON TEST IMAGES WITHOUT IMAGE PERTURBATIONS

Classifier	Accuracy
Discriminative	98.0%
Generative	97.9%

The test accuracies of the discriminative and generative classifiers are shown in Table II. The confusion matrix for the test of the discriminative classifier is shown in Fig. 9. The confusion matrix for the test of the generative classifier is shown Fig. 10. The confusion matrices proves that both classifiers were least accurate on nines, which were most commonly confused with fours for both classifiers. Also, the confusion matrices showed that both classifiers confused twos and sevens. The confusion matrix for the generative classifier also shows confusion of threes and fives.

C. Accuracy on Perturbed Test Images

Performance of the classifiers when test images were perturbed by FGSM is shown in Fig. 11, which shows that the generative classifier was more robust against the FGSM

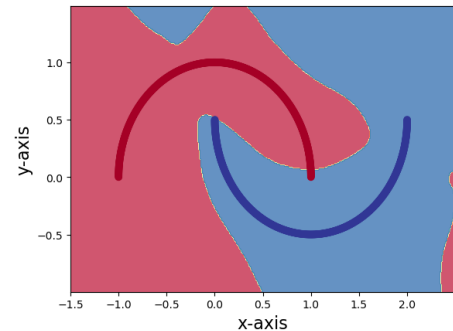


Fig. 5. Two moons generative classifier modeling the classes locations on test data points without perturbations

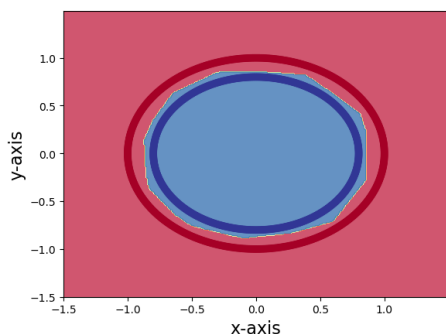


Fig. 6. Decision boundary drawn by the two circles discriminative classifier on test data points without perturbations

attack than the discriminative classifier. Performance of the classifiers, trained without image perturbations, when test images were perturbed with the stochastic noises are shown using \circ markers in Fig. 12, Fig. 13 and Fig. 14. The plots also shows the performance of classifiers trained with perturbations using $*$ markers, which will be discussed later. The plots show that the discriminative classifier was more robust than the generative classifier against stochastic image perturbations, when trained on images free from perturbations.

D. Results of Training Classifiers with Image Perturbations

The accuracy of classifiers (trained with image perturbations) on perturbed images are shown with $*$ markers in Fig. 12, Fig. 13 and Fig. 14. The discriminative classifier became more robust when some of the training images were perturbed. The generative classifier was more accurate for large perturbations but less accurate for small perturbations when some training images were perturbed with Gaussian noise and negative impulse noise. However, the generative model trained on impulse noise was not able to generalize. The performance of the generative classifier trained with image perturbations is explained under section VI.

VI. DISCUSSION

A. Performance of Toy Data Classifiers

The discriminative and generative classifiers classify differently, but the result was similar. The discriminative classifier finds separating characteristics and draws a boundary between

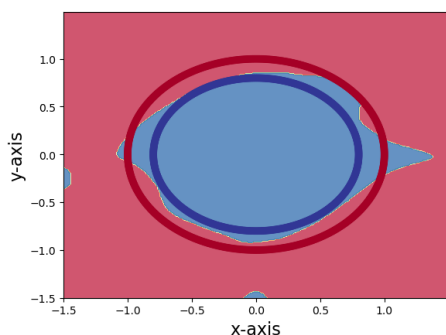


Fig. 7. Two circles generative classifier modeling the classes locations on test data points without perturbations

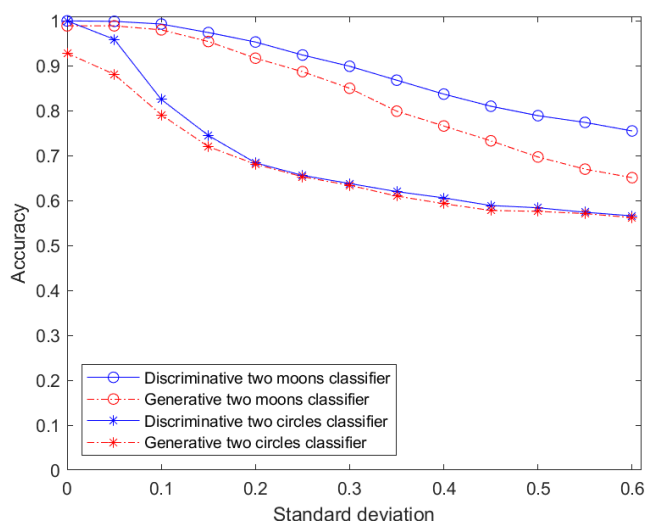


Fig. 8. Accuracy of the discriminative and generative classifiers on toy datasets plotted against the standard deviation of Gaussian noise added to the test data

the classes. The generative classifier tries to learn the characteristics of each cluster to predict which cluster would be most likely to have produced a data point.

The classified regions by the classifiers on two moons were different. When Gaussian noise perturbed two moons the discriminative classifier was the most accurate for all standard deviations of the Gaussian noise. The generative classifier gave a lot of the area very far away from the clusters, which was probably why the discriminative classifier was more robust against Gaussian noise.

The classified regions by the classifiers on two circles were very similar. This could explain the similar performance when data was perturbed with Gaussian noise. The discriminative classifier had a bit better accuracy on unperturbed data than the generative classifier. However, for data perturbed with Gaussian noise with standard deviation greater or equal to 0.2

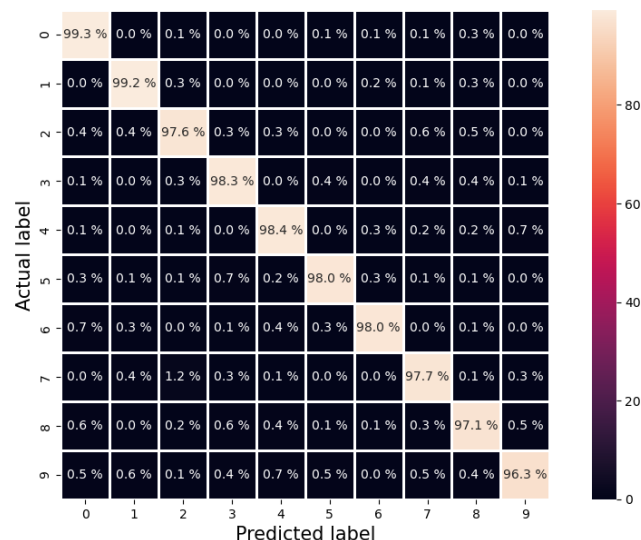


Fig. 9. Confusion matrix for discriminative classifier on test images free from image perturbations

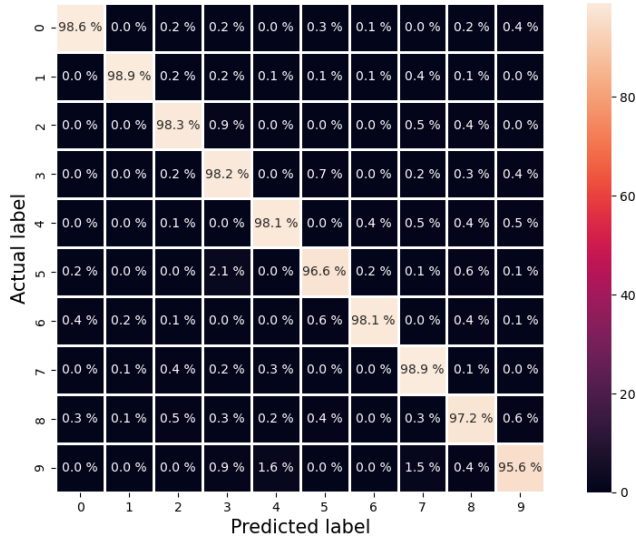


Fig. 10. Confusion matrix for generative classifier on test images free from image perturbations

the accuracy was similar.

Outliers could cause trouble for the generative classifier to decide which class could have generated the data point. However, the discriminative classifiers decision boundary could handle many extreme outliers and classify them correctly. Which could explain why the discriminative classifier was always more accurate than the generative classifier.

B. The Generative Image Classifiers' Autoencoder

The autoencoder enabled the generative model to learn a less complex mapping, which made the classifier less computationally expensive. Still, the generative classifier was about ten times as computationally expensive as the discriminative classifier. Since the generative classifier used an autoencoder it was not entirely based on normalizing flows (because the autoencoder was constructed using a CNN). Some of the results presented in this report could possibly be hindered

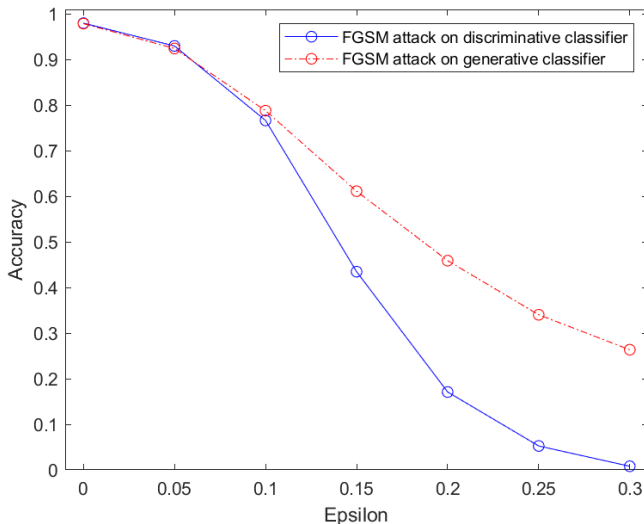


Fig. 11. Accuracy of the discriminative and generative classifiers attacked with FGSM for different weights ϵ of perturbation applied to the test images

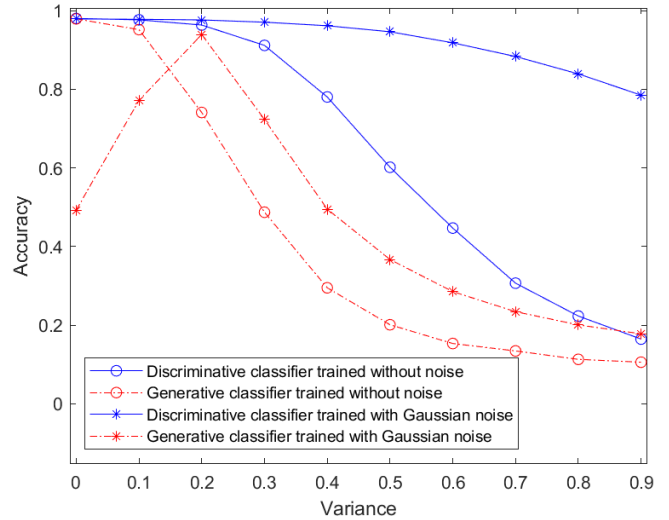


Fig. 12. Accuracy of discriminative and generative classifiers, trained with and without Gaussian noise, on images perturbed with Gaussian noise plotted against variance of the Gaussian noise

to be generalized for models implemented completely in normalizing flows. To ensure the results from this project are valid for other generative classifiers more studies are needed.

C. Performance of Image Classifiers on Unperturbed Images

The performance of the classifiers were very similar on unperturbed images and both classifiers generalized very well on unseen images without perturbations. This could lead to the conclusion that the model learns from what has been encountered in training and explains the decrease in accuracy when images were perturbed (the perturbed images have not previously been seen in training). Because the total accuracy for the two classifiers was 98%, there was very little confusion and therefore it is difficult to draw any conclusions about misclassification based on the confusion matrices.

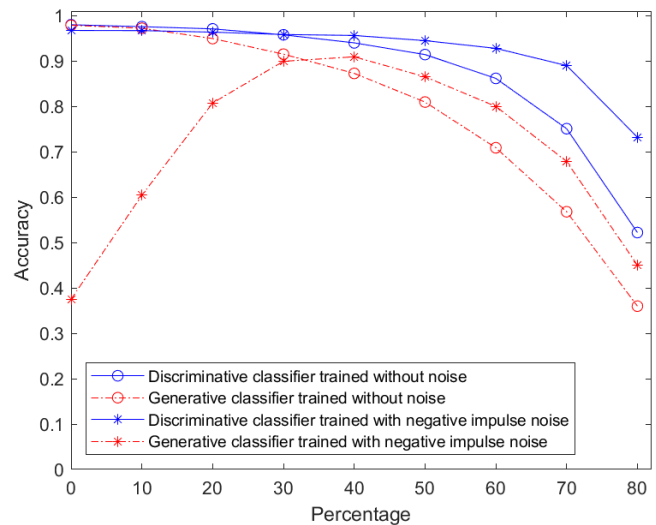


Fig. 13. Accuracy of discriminative and generative classifiers, trained with and without negative impulse noise, on test images perturbed with negative impulse noise plotted against percentage of total pixels affected by negative impulse noise

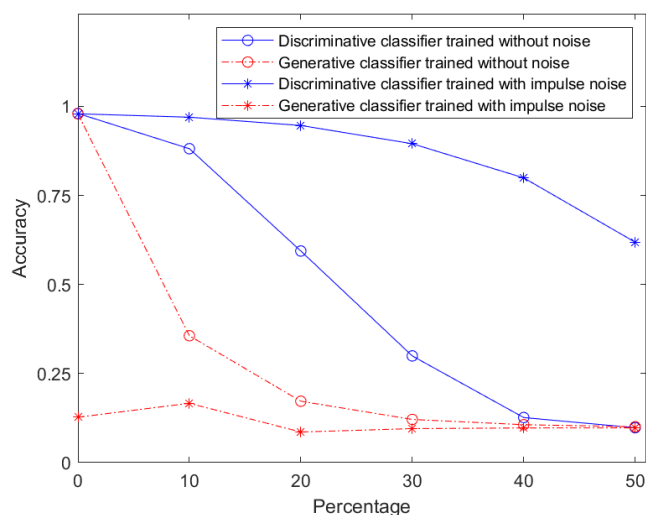


Fig. 14. Accuracy of discriminative and generative classifiers, trained with and without impulse noise, on test images perturbed with impulse noise plotted against percentage of total pixels affected by impulse noise

D. Image Classifiers Robustness Against FGSM Attack

The generative classifier was more robust than the discriminative classifier against the FGSM attack, especially for large perturbations. The perturbations caused by the FGSM attack were a bit different for the two classifiers, displayed in Fig. 2. The attack on the discriminative classifier was less specific and changed pixels all over the image. The attack on the generative classifier targeted pixels close to the boundary of the digit. Images perturbed with respect to the discriminative classifier were significantly more perturbed than the images perturbed with respect to the generative classifier. This could be explained by the fact that the discriminative classifier used kernels to extract features. To make it harder for the feature extraction (or cause wrong features to be extracted) pixels all over the image were manipulated. The generative classifier tried to sense which class could have generated the image. All the images used in training had white pixels in the middle of the image and the boundaries of the image contained black pixels. Therefore the attack on the generative classifier only changed pixels in the middle of the image, because no class would have generated coloured pixels at the edges of the image.

The FGSM attacks were performed with the same method on the classifiers and the generative classifier was more robust against the attack. The robustness of the generative classifier could therefore be explained by the fact that it was more difficult to systematically find an effective FGSM attack than for the discriminative classifier.

E. Image Classifiers Robustness Against Stochastic Image Perturbations

The discriminative classifier was significantly more robust against image perturbations with Gaussian, positive impulse and negative impulse noise. The discriminative classifier had a higher accuracy at all noise levels than the generative classifier. This could be explained by the way the classifiers classified data. The discriminative classifier determined a boundary

between the classes based on feature extraction from the images. The generative classifier estimates which class could have produced the image by learning the characteristics of the different classes. The decision boundary determined by the discriminative classifier was slightly different for perturbed images than unperturbed images. However, the slightly different decision boundary managed to place many outliers within the correct decision boundaries. Meaning that the feature extraction still managed to extract relevant features from the perturbed images. On the other hand, the generative classifier struggled to determine which class could have generated the images containing stochastic image perturbations. Because images seen in training were free from noise it is understandable why the generative classifier struggled to sense which image class could have generated the perturbed images. This resulted in the discriminative classifier being more accurate than the generative classifier on images perturbed with Gaussian, positive impulse and negative impulse noise.

F. Influence of Training Image Classifiers With Image Perturbations

The discriminative classifier became more robust against perturbations when trained on a mix of unperturbed and perturbed images. The reason why the discriminative classifier got more robust when trained on noise was that the added noise was stochastic, hence all the perturbed images were different and simply made the training set larger. The training set was expanded with outliers of the handwritten numbers, which also could have helped the classifier to generalize well on perturbed images. Furthermore, adding perturbed images also reduced the risk of overfitting. This leads to the conclusion that adding some perturbed images in training of the discriminative classifier made the classifier more robust against perturbations seen in training.

The discriminative classifier, trained with perturbations, was noticeably accurate on images with large perturbations, which are displayed in Fig. 15. Images containing Gaussian noise with variance 0.9 would be almost impossible for a

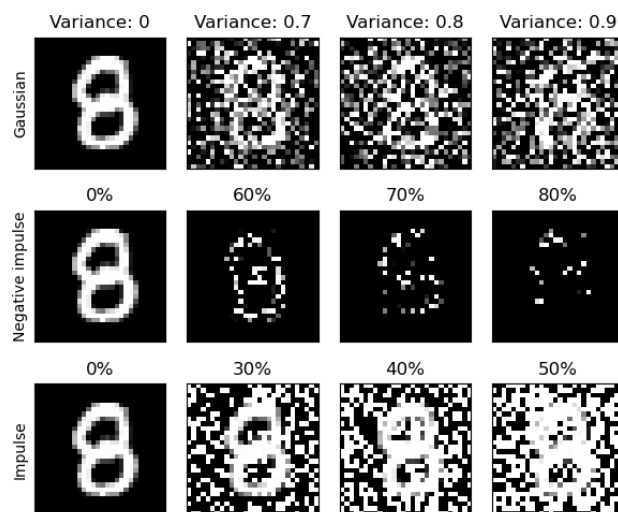


Fig. 15. Visualization of large stochastic noise perturbations

human to classify correctly consistently, but the discriminative classifier (trained with Gaussian noise) had an accuracy of 80%. Furthermore, images where 80% of pixels have been perturbed with negative impulse noise could be difficult for a human to classify, but the discriminative classifier (trained with negative impulse noise) had an accuracy above 70%. Lastly, images where 50% of pixels have been perturbed with positive impulse noise would be challenging for a human to classify, but the discriminative classifier (trained with positive impulse noise) had an accuracy above 60%. These three results could lead to the conclusion that a discriminative classifier (trained on a mix of unperturbed and perturbed images) could classify images with large perturbations, of which most humans would struggle to classify. It can also be concluded that the feature extraction works well on stochastic image perturbations.

The result of training the generative classifier with image perturbations was a bit different than for the discriminative classifier. A hypothesis for the result is that the generative classifier learned the mean structure of the training images. When trained with Gaussian noise half the train images were unperturbed and the other half contained Gaussian noise with variance 0.5. The mean structure from training should therefore have Gaussian noise with variance 0.25, which could explain why the classifier's measured peak accuracy was on images containing Gaussian noise with variance 0.2. When the generative classifier was trained with negative impulse noise, half of the images used in training were perturbed by adding negative impulse noise to 70% of the pixels and the other half was unaffected by perturbations. The mean structure learned by the classifier would therefore have 35% of pixels perturbed with negative impulse noise. This could explain the fact that the classifier's measured peak accuracy was on images with 40% of pixels perturbed with negative impulse noise. For the generative classifier trained with positive impulse noise the mean structure should have been at 5% and the measured peak was on images where 10% of the pixels was perturbed with positive impulse noise. However, it is hard to draw any conclusions about the learning of the classifier trained with positive impulse noise because the accuracy was very close to 10%, which was the accuracy the classifier would have had if it classified images randomly. None of these results does argue against the formulated hypothesis, however since the accuracy was not measured at the theoretical peak accuracy (which could verify this hypothesis) the results does not confirm the hypothesis either. In order to confirm the hypothesis, more studies are needed.

The generative model trained on positive impulse noise did not manage to generalize. This was due to underfitting, the classifier did not learn to model the training images. This was verified by classifying images used in training with the trained classifier and the accuracy was 13%. It can therefore be concluded that the generative architecture was able to model unperturbed images, Gaussian noise perturbed images and negative impulse noise perturbed images. However, the architecture was not sufficient to model (positive) impulse noise perturbed images. In future studies the underfitting problem could be solved by creating a more complex architecture that manages to model the positive impulse noise in training.

G. Future Studies

In future studies, more extensive comparisons could be made between discriminative and generative classifiers. One future study could be to implement a generative classifier without the autoencoder (map directly from an image to a simple probability distribution) and compare the performance to the generative classifier used in this study. A classifier without the autoencoder would be significantly more computationally expensive. However, the classifier could potentially be more robust against perturbations. This was due to the autoencoder's mapping from image to embedding being the most noise sensitive part of the generative classifier. It would also be of interest to study different generative classifiers, such as generative adversarial networks, to see if they perform differently than the generative classifier in this report.

In this report the discriminative classifier exhibited a significant robustness against image perturbations, when some of the images in training were perturbed. In future work, studies could be made to investigate if the classifier could be trained to be more robust against adversarial examples.

The results of introducing perturbed images in training of the generative classifier suggested that the classifier learnt the mean structure of the image classes. This hypothesis could neither be debunked or verified with the results in this report. In future work, the hypothesis could be tested by verifying that the classifier performs the best on images containing the average perturbation used in training, for different averages.

Another topic that could be investigated is the explainability of the classifiers. This could be examined by visualizing activations in the convolutional layers and flows of the flow layers. Perhaps it would be possible to see which features the classifiers learn to distinguish classes.

VII. CONCLUSIONS

The conclusions drawn in this report can be summarized:

- The discriminative and generative image classifiers had similar performance on unperturbed test images
- The generative classifier was more robust against the FGSM attack than the discriminative classifier
- The discriminative classifier was more robust against Gaussian image perturbations and impulse image perturbations than the generative classifier
- The discriminative classifier was significantly more robust against image perturbations that was added to some images in training

ACKNOWLEDGMENT

The authors would like to express our sincere gratitude to our supervisor Anubhab Ghosh for his patience, support and friendly guidance.

REFERENCES

- [1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge: MIT Press, 2016.
- [2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, Jan, 2015. [Online]. Available: <https://arxiv.org/abs/1409.0575>

- [3] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," *arXiv:1412.6572v3*, Mar, 2015. [Online]. Available: <https://arxiv.org/abs/1412.6572>
- [4] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do ImageNet Classifiers Generalize to ImageNet?," in *Proceedings of the 36th International Conference on Machine Learning*, 2019, pp. 5389–5400. [Online]. Available: <https://proceedings.mlr.press/v97/recht19a.html>
- [5] I. Kobyzev, S. J. Prince, and M. A. Brubaker, "Normalizing Flows: An Introduction and Review of Current Methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 11, pp. 3964–3979, Nov, 2021. [Online]. Available: <http://dx.doi.org/10.1109/TPAMI.2020.2992934>
- [6] Y. Li, J. Bradshaw, and Y. Sharma, "Are Generative Classifiers More Robust to Adversarial Attacks?," in *Proceedings of the 36th International Conference on Machine Learning*, 2019, pp. 3804–3814. [Online]. Available: <http://proceedings.mlr.press/v97/li19a/li19a.pdf>
- [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, Nov, 1998. [Online]. Available: <https://ieeexplore.ieee.org/document/726791/keywords#keywords>
- [8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, A. Müller, J. Nothman, G. Louppe, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and D. Duchesnay, "Scikit-learn: Machine Learning in Python," *arXiv:1201.0490v4*, Jun, 2018. [Online]. Available: <https://arxiv.org/abs/1201.0490>
- [9] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling, "Semi-supervised Learning with Deep Generative Models," in *Advances in Neural Information Processing Systems*, vol. 27, 2014. [Online]. Available: <https://proceedings.neurips.cc/paper/2014/file/d523773c6b194f37b938d340d5d02232-Paper.pdf>
- [10] L. Bruzzone and D. Prieto, "Unsupervised retraining of a maximum likelihood classifier for the analysis of multitemporal remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 2, pp. 456–460, Feb, 2001. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/905255>
- [11] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of trends in Practice and Research for Deep Learning," *arXiv:1811.03378v1*, Nov, 2018. [Online]. Available: <https://arxiv.org/abs/1811.03378>
- [12] A. F. Agarap, "Deep Learning using Rectified Linear Units (ReLU)," *arXiv:1803.08375v2*, Feb, 2019. [Online]. Available: <https://arxiv.org/abs/1803.08375>
- [13] S. Narayan, "The generalized sigmoid activation function: Competitive supervised learning," *Information Sciences*, vol. 99, no. 1, pp. 69–82, Jun, 1997. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0020025596002009>
- [14] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for Simplicity: The All Convolutional Net," *arXiv:1412.6806v3*, pp. 1–2, Apr, 2015. [Online]. Available: <https://arxiv.org/pdf/1412.6806.pdf>
- [15] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, "Dive into Deep Learning," *arXiv:2106.11342*, Jul, 2021. [Online]. Available: <https://arxiv.org/abs/2106.11342>
- [16] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using Real NVP," *arXiv:1605.08803v3*, Feb, 2017. [Online]. Available: <https://arxiv.org/abs/1605.08803>
- [17] L. Li, M. Doroslovački, and M. H. Loew, "Approximating the Gradient of Cross-Entropy Loss Function," *IEEE Access*, vol. 8, pp. 111 626–111 635, Jun, 2020. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9113308>
- [18] L. Bottou, "Large-Scale Machine Learning with Stochastic Gradient Descent," in *Proceedings of COMPSTAT'2010*, 2010, pp. 177–186. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-7908-2604-3_16#citeas
- [19] A. Ashukha. (2018, Aug.) Real NVP PyTorch. GitHub. [Online]. Available: <https://github.com/senya-ashukha/real-nvp-pytorch/blob/master/real-nvp-pytorch.ipynb>
- [20] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier Nonlinearities Improve Neural Network Acoustic Models," in *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013. [Online]. Available: https://ai.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf
- [21] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980v9*, Jan, 2017. [Online]. Available: <https://arxiv.org/abs/1412.6980>
- [22] (2021, May.) PyTorch Convolutional Neural Network With MNIST Dataset. Medium. [Online]. Available: <https://medium.com/@nutanbhogendrasharma/pytorch-convolutional-neural-network-with-mnist-dataset-4e8a4265e118>
- [23] S. Scardapane and J. Pomponi. (2020, Mar.) RealNVP on MNIST. GitHub. [Online]. Available: <https://github.com/ispamm/realnvp-demo-pytorch>
- [24] U. Ruby and V. Yendapalli, "Binary cross entropy with deep learning technique for Image classification," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 10, Oct, 2020. [Online]. Available: https://www.researchgate.net/publication/344854379_Binary_cross_entropy_with_deep_learning_technique_for_Image_classification
- [25] H. Yao, D. Zhu, B. Jiang, and P. Yu, "Negative Log Likelihood Ratio Loss for Deep Neural Network Classification," in *Proceedings of the Future Technologies Conference*, 2019, pp. 276–282. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-32520-6_22
- [26] M. Jayamanmadharao, S. Anuradha, and K. Reddy, "Impulse Noise removal in Digital Images," *International Journal on Computer Science and Engineering*, vol. 2, Oct, 2010. [Online]. Available: https://www.researchgate.net/publication/50194206_Impulse_Noise_removal_in_Digital_Images
- [27] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," *arXiv:1912.01703v1*, Dec, 2019. [Online]. Available: <https://arxiv.org/abs/1912.01703>

The Impact of Noise on Generative and Discriminative Image Classifiers

Maximilian Stenlund and Valdemar Jakobsson

Abstract—This report analyzes the difference between discriminative and generative image classifiers when tested on noise. The generative classifier was a maximum-likelihood based classifier using a normalizing flow as the generative model. In this work, a coupling flow such as RealNVP was used. For the discriminative classifier a convolutional network was implemented. A detailed description of how these classifiers were implemented is given in the report. The report shows how this generative classifier outperforms the discriminative classifier when tested on adversarial noise. However, tests are also conducted on salt and pepper noise and Gaussian noise, here the results show that the generative classifier gets outperformed by the discriminative classifier. Tests were also conducted on Gaussian noise once both classifiers had been trained on Gaussian noise, the results from these tests show that the discriminative classifier performs significantly better once trained on Gaussian noise. However, the generative classifier does only show marginal increases in performance and performs worse on clean data once trained on Gaussian noise.

Sammanfattning—Den här rapporten analyserar skillnaden mellan diskriminativa och generativa modellklasser för bildigenkänning när de testas på brus. Den generativa modellklassen var en maximum-likelihood baserad generativ klassifikationsmodell. Inom detta arbete användes kopplingsflödet RealNVP. För den diskriminativa bildigenkänningsmodellen så implementerades ett faltningsnätverk. En detaljerad beskrivning för hur dessa bildigenkänningsmodeller genomfördes är given i rapporten. Rapporten visar hur den generativa modellklassen överträffar den diskriminativa modellklassen när de testas på adversariellt brus. Testerna utförs emellertid med salt och peppar brus och Gaussiskt brus, för dessa visar resultaten att den generativa modellklassen överträffas av den diskriminativa modellklassen. Den generativa modellklassen visar emellertid endast marginella öknings i prestanda, och har en sämre prestanda på ren data efter att den tränats på Gaussiskt brus.

Index Terms—Artificial intelligence, Adversarial noise, Discriminative, Generative, Salt and Pepper noise, Gaussian noise, neural networks, Normalized flows, Convolutional networks

Supervisors: Anubhab Ghosh and Saikat Chatterjee

TRITA number: TRITA-EECS-EX-2022:174

I. INTRODUCTION

A. Background

Convolutional neural networks (CNNs) are useful and popular when it comes to computer vision tasks, but they have a fatal flaw. Due to the huge amount of parameters in the network, the CNNs have the risk of overfitting. It is possible to conclude that by training CNNs on different types of noise a more robust feature representation can be achieved and the risk of overfitting would therefore be reduced according to [1].

One type of noise that is particularly interesting to study is adversarial noise. Small modifications can be made to the input data of a high-performing network that will make the network misclassify every example. When these modifications are applied to an image dataset the change in the images can often not be detected by the human eye. The impact of adversarial noise on neural networks introduces potential vulnerabilities when used in practical situations. An example of such a vulnerability would be if an attacker had knowledge of the models parameters and attacked the network with adversarial examples to make the network fail. The example shows that there is a big gap between the robustness of neural networks and human perception even though the gap has seen a huge reduction in recent years as is explained in [2].

There are two types of image classifiers, generative and discriminative classifiers. A discriminative image classifier learns the differences between each class of images and then defines a boundary between each class. After the discriminative classifier is trained it can then give a probability that a certain image is within each class of images in order to classify an image. The generative classifier instead creates a model for each class of images and then defines a boundary around each class of images. It is then capable of not only returning the probability that an image is within the corresponding model, but it is also able to generate images of said class of images. Identifying which of these two classifiers performs best on adversarial noise is the main focus of this report, and will conclusively show which classifier is more robust.

II. PROBLEM FORMULATION

The main goal of the thesis is to implement a test environment in order to test a generative and a discriminative image classifier on a set of data that has been distorted with some type of noise. The thesis is mostly concerned with the difference between these two classifiers for data which has been perturbed with adversarial noise as the input to the classifier. However, the impact of Gaussian noise, as well as salt and pepper noise will also be analyzed in this report. The dataset which is used to both train and validate these classifiers is the MNIST dataset, which is a dataset consisting of 60000 hand-drawn digits from 0 to 9. Further information about the MNIST dataset is given in [3].

III. THEORY

In this section a thorough description of the theory behind each part of the project is given. First, an explanation is given

for how the different classifiers works, then a description for each type of noise is given.

A. Neural network

The interest of this article relates to the differences in both generative and discriminative classifiers, in order to understand such classifiers the concept of a neural network requires explanation. As is explained in [4] a neural network is a collection of connected nodes in a graph. The signal given at a connection of such a node is a real number which the node then applies to some non-linear transformation, in order to give an output for the next node in the network.

There is a multitude of different types of neural networks that can be applied to certain problems. When it comes to the discriminative classifier, this article is concerned with so-called convolutional networks. As is explained in [5] a convolutional network is a special kind of neural network which is mainly applied to process data that has a topology similar to a grid. An example of data that can be stored as a grid is an image. A layer in a convolutional network typically consists of three stages. First, several convolutions are applied in parallel in order to produce a set of linear activations in the nodes, then each linear activation is sent through some non-linear activation function, then a pooling function is used to modify the output of the layer. As is mentioned in [5] the pooling function replaces the output of the network at a location with a summary of the nearby outputs. Hence, the pooling function will make the convolutional network invariant to small changes in the input. For convolutional networks the pooling function is typically used for regularization.

When training a neural network a training set is typically used. The training of the neural network is done by minimizing training error using a cost function [5]. However, this is not the only error that should be minimized. There is also an error called the test error which is defined as the expected error when the neural network is given new input. When the network tries to minimize these errors there are two corresponding challenges that arise; underfitting and overfitting. Underfitting is when the error on the training set isn't low enough. Overfitting on the other hand is when the gap between the training error and the test error is too large. The way to control if the model under- or overfits is by controlling the capacity. The model will perform the best when the capacity matches the complexity of the task that should be performed and the amount of data that is available. If the capacity is too low the model can't solve complex tasks. If the capacity is too high the model will be able to match the data points exactly but this is not good either since there can be infinitely many solutions that fit the data points so there is a low probability to choose an adequate solution.

B. Discriminative classifier

The assignment of the discriminative classifier is to identify the class membership of y given unknown data x

in a dataset $D = (x_1, y_1) \dots (y_n, x_n)$ where x_i has known class membership y_i according to [6]. Since there usually isn't a functional relationship $y = f(x)$ between x and y the relationship is described more generally by the probability distribution $p(x, y)$, and the class label y should be chosen to maximize the posterior distribution $p(x|y)$. To classify the MNIST dataset the y labels are 0-9 and x_i is a $28 * 28$ -dimensional matrix.

For the discriminative classifier in this report, a convolutional neural network will be used. Due to the nonlinearity in the hidden neurons of the neural network, the output will be a non-linear function of the inputs which means that the decision boundary between the class labels can be non-linear as well as reported by [6].

The model parameters for the discriminative classifier are chosen by maximum-likelihood estimation. The idea behind maximum likelihood estimation is to asses the parameter 'a' so that the measured data becomes as likely as possible. Let $x_1 \dots x_n$ be the outcome of the stochastic variables $X_1 \dots X_n$. The probability to get the given outcome is $P(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n)$. This probability should be maximized which means maximizing equation 1

$$\prod_{i=1}^n p(y_i | x_i, a) \quad (1)$$

if independent and identically distributed random variables are assumed as is stated in [6]. To make it easier equation 2

$$- \sum_{i=1}^n \log(p(y_i | x_i, a)) \quad (2)$$

can often be minimized instead since that is often equivalent. There are many different numerical optimization algorithms that can be used to determine the parameters. In this report, gradient descent will be used.

To train a discriminative classifier an error function is often used. An appropriate choice for such a function is the cross-entropy error which is given by equation 3

$$\sum_{i=1}^n y \log(o_n) + (1 - y) \log(1 - o_n) \quad (3)$$

where o_n is the output of the network according to [6]. The linear connection layer of the perceptron can be represented as $Y = W_n W_{n-1} \dots W_1 X$ and for the full connection layer it can be expressed as the composite function $Y = f_n(f_{n-1}(\dots f_1(X) \dots))$.

In figure 1 the discriminative classifier has been trained on two-dimensional data x with the class labels 0 and 1 for y . The decision boundary is colored white and can be seen separating the two moon-like structures. The dataset illustrating this moon-like structure can be found in [7].

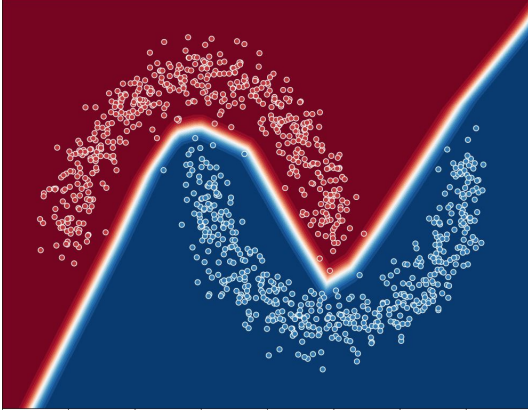


Fig. 1. Decision boundary drawn by the discriminative classifier

C. Gaussian mixture models

There are two types of image classifiers, one of them is the previously mentioned discriminative classifier, the other is a so-called generative classifier. This type of classifier creates one model for each class of images, contrary to the discriminative classifier which has one model for all classifiers.

One type of these generative classifiers is the GMM (Gaussian Mixture Models). As is mentioned in [8] a GMM is the weighted sum of all Gaussian densities as is described in equation 4.

$$p(x|\lambda) = \sum_{i=1}^M w_i \mathcal{N}(x|\mu_i, \Sigma_i) \quad (4)$$

Where M is the number of classifications, x is a D -dimensional continuous-valued data vector for this case it's the images that should be classified by the classifier, w are the weights of the classifier, \mathcal{N} are the components Gaussian densities. Each density is a D -variate Gaussian function of the form described in equation 5.

$$\mathcal{N}(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{D/2} |\Sigma_i|} \exp\left(-\frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)\right) \quad (5)$$

μ_i is the mean vector and Σ_i is the covariance matrix. The weights w_i satisfy the condition that $\sum_{i=1}^M w_i = 1$.

In order to make a prediction as to which class a given image is within, one can create a maximum likelihood algorithm for the GMM. This implies that one has to retrieve the probability for each model and then compare the probabilities, the model which returns the greatest probability is then the prediction that the GMM gives. For each model i one can retrieve the probability for the model that the given image is within via using equation 6:

$$Pr(i|x_i, \lambda) = \frac{w_i \mathcal{N}(x|\mu_i, \Sigma_i)}{\sum_{k=1}^M w_k \mathcal{N}(x|\mu_k, \Sigma_k)} \quad (6)$$

D. Normalized flow

For this project a more sophisticated generative classifier was applied, namely normalized flow. For normalized flow, one can let some $Z \in R^D$ be a random variable with know probability density function p_Z . Let $Y = g(Z)$ be a bijective function and $f(Y)$ be the inverse of $g(Z)$, then the formula given in equation 7 for normalized flow is given, as is mentioned in [9]:

$$p_Y(y) = p_Z(f(y)) |det(\mathcal{D}f(y))| \quad (7)$$

Where $\mathcal{D}f(y)$ denotes the Jacobian of $f(y)$. This probability density function $p_Y(y)$ is given the name pushforward of the density p_Z . In the context of this article, the function g is the generator and it pushes the prior density p_Z to a more complex density. Here one could note that due to the nature of g , one can generate new data based on the data that the model is trained on. Such generated images can be seen in figure 2.



Fig. 2. Images generated by the normalized flow models trained on the MNIST dataset

The function $f(y)$ flows in the normalizing direction and normalizes the data distribution. Where the normalizing direction is the opposite direction of the generative direction, which is the direction that moves the base density p_Z to the final more complicated density. Hence the name normalizing flows. It is also important to mention that a flow is often modeled in the normalized direction, due to the inverse often being difficult to compute as is mentioned in [9]. In order to achieve the likelihood that a set of data D with M data points y is given by parameters $\beta = (\theta, \phi)$, can be calculated from equation 8 as is also mentioned by [9].

$$\log(p(D|\beta)) = \sum_{i=1}^M (\log(p_Z(f(y^{(i)}|\theta))|\phi)) + \log|det \mathcal{D}f(y^i|\theta)| \quad (8)$$

During the training of the normalized flow, θ (the parameters of the flow) and ϕ (the parameters of the given base distribution) are optimized to maximize this log-likelihood. After the training is complete a maximum likelihood algorithm can be applied to equation 8 in order to determine which parameters the input data points belong to, and henceforth one also receives the class of the data points.

There is a multitude of different methods to apply normalized flows. However, in this article the main focus will be on RealNVP, which is a coupling flow. To understand coupling flows one could consider the input $x \in R^D$ and partition it into two subspaces $x^A \in R^d$, $x^B \in R^{D-d}$ as in [9]. Then a bijection $h(\cdot; \theta) : R^d \rightarrow R^d$ can be defined, parameterized by θ . From this information, one can define the following equations 9 and 10 for the coupling flow.

$$y^A = h(x^A; \Theta(x^B)) \quad (9)$$

$$y^B = x^B \quad (10)$$

In equation 9 the function $\Theta(x^B)$ is any arbitrary function that can only have x^B as input, this function also defines the parameters θ for the classifier and is called a conditioner as is mentioned in [9]. Equations 9 and 10 is what define the coupling flow g . For a graphical illustration see figure 3. The Jacobian given by g is a triangular matrix where the diagonals are $\mathcal{D}h$ and the identity matrix, hence the reasoning behind using a coupling flow, as the Jacobian is faster to compute. In order to apply the coupling flow to some input z a block diagram is given in figure 4.

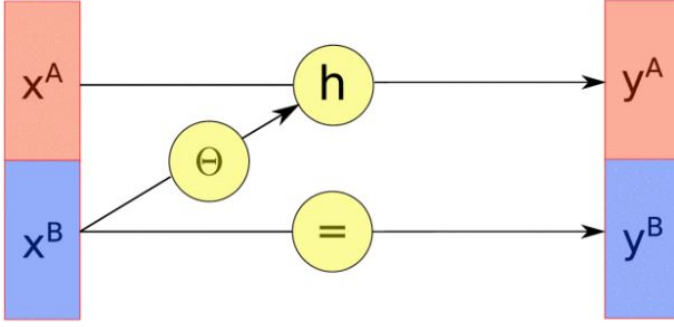


Fig. 3. Block diagram defining the function g [9]

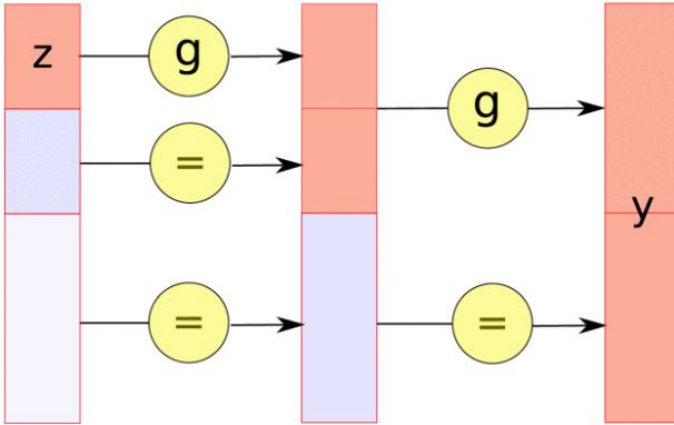


Fig. 4. Block diagram of the function g applied on z [9]

In figures 5 and 6, the generation of the datapoints by $g(Z)$ is described via the use of a simple dataset, where figure 5 describes the input on which the dataset is trained, and figure 6 is the datapoints generated by $g(Z)$. The boundary that is given in figure 6 is the decision boundary of the model, in this case, two RealNVP models have been created 1 for each 'half moon' in the figure. The decision boundary drawn by the model corresponding to the lower moon is blue, and the decision boundary for the upper moon is displayed in red, overlapping of the two models decision boundaries is shown in purple. This simple dataset is the same dataset on which the discriminative classifier was trained.

E. Adversarial noise

Adversarial noise is the main topic of this article, it's simply noise that is applied by exploiting information that is obtainable via the model. Hence, it does not necessarily

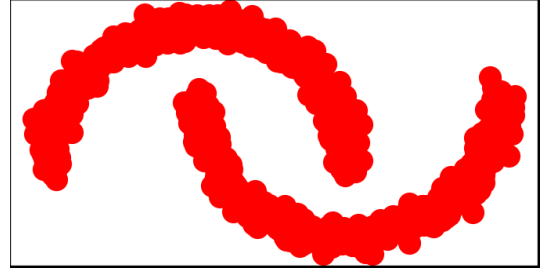


Fig. 5. Input to the generative classifier

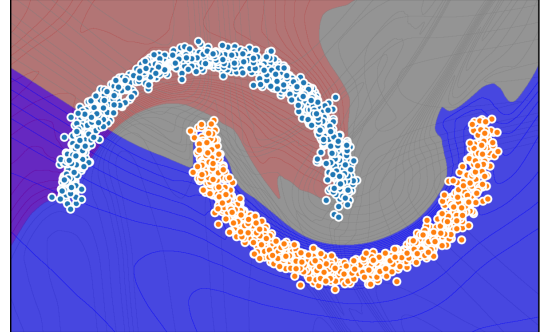


Fig. 6. Decision boundary drawn by the generative classifier

distort the information to the point that it's impossible to understand the information for a human. However, the main idea behind adversarial noise lies in distorting the information given to the point where a model can't correctly analyze the information.

The methodology used to apply adversarial noise to a discriminative classifier is generally known as the Fast Gradient Sign Method (FGSM). The equation describing how the method is applied to distort some image x is:

$$\hat{x} = x + \epsilon \text{sign}(\nabla_x J(\theta, x, y)) \quad (11)$$

In equation 11, ϵ is the amount of desired distortion that will be applied to the image, x is the image, y is the label of the image, J is the loss function and θ is the parameters for the model (bias and weights). Hence in order to generate such noise, one is required to calculate the gradient of the loss function given the label of the image that the noise shall be applied to. In figures 7 and 8 one can see two images that have been generated using this method using a generative and a discriminative image classifier.

F. Gaussian noise

Gaussian noise is a rather different way to apply noise compared to adversarial noise. In the case of generating Gaussian noise on an image, knowledge about the model is not used. The noise generated by Gaussian noise is equal to that of the probability density function of a normal distribution, which can be seen in equation 12, and is also seen in [10].

$$p_G(z; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \quad (12)$$

This can then be used to distort images in order to identify how well an image classifier can handle such distortion. A distorted image where $\sigma = 0.5$ and $\mu = 0$ can be seen in figure 9.

G. Salt and pepper noise

Salt and pepper noise can be caused by a wide variety of sources and what they all have in common is that they make only some of the pixels in the image noisy. However, the pixels that get noisy get extremely noisy to the point where they are either white or black. The effect of the salt and pepper noise will make the picture look as if salt and pepper are sprinkled over the image. An example of when this could happen is when images are transmitted over noisy digital links according to [11].

To describe what is happening to the image the following model could be used. If the original image corresponds to $f(x, y)$ and the picture that has been altered corresponds to $g(x, y)$ then the model can be described by equations 13, 14 and 15:

$$P(g = f) = 1 - \alpha \quad (13)$$

$$P(g = \max) = 1 - \alpha/2 \quad (14)$$

$$P(g = \min) = 1 - \alpha/2 \quad (15)$$

. In these equations, the max value is the maximum value of a pixel which corresponds to a 1 in this report and means that the pixels are white. The min corresponds to the minimum value of a pixel which for this report is a 0 and means that the pixel is black. Alpha is a probability that varies depending on how noisy the image is. A higher value of alpha corresponds to a noisier image and if the value of alpha is 1, all of the pixels in the image will be noisy. The same model can be found in [11].

To see an example of how a picture in the MNIST data set can look with an alpha of 0.2 see figure 10.

IV. METHOD

In this section a description of how each classifier was implemented is given. The methodology for implementing the fast gradient sign method in order to generate adversarial noise is also given a description here. Implementations of the following methodologies was done using PyTorch [12].

A. Discriminative

To implement the discriminative classifier the open-source machine learning framework PyTorch was used. The convolutional neural network was modeled by building a class inheriting from the torch.nn module. The network consists of two convolutional parts with kernel size 5*5 and with the first part having 1 channel as input and 6 channels as outputs and the second part having 6 channels as input and 16 channels as output, implemented with torch.nn.Conv2d(). The convolutional part of the network is connected to a neural network with an input size of 256, two hidden layers

with sizes 120 and 84, and an output layer of size 10 that are implemented using torch.nn.Linear().

The classifier was then trained on the MNIST dataset. The optimiser that was used was stochastic gradient descent from torch.optim.SGD() and the loss function was modelled using cross entropy loss from torch.nn.CrossEntropyLoss(). The total numbers of epochs for the training of this classifier was 30 and the learning rate for the optimiser was set to 10^{-2} .

B. Generative

For the generative image classifier the coupling flow RealNVP was used. However, in order to train the RealNVP classifier efficiently an Autoencoder was also trained in order to use the outputs from the Autoencoder to train the RealNVP. The Autoencoder as well as the RealNVP model was taken from [13]. Slight changes was made to the code for the RealNVP model, as the model given by [13] is technically not a generative classifier since it only returns 1 model. Hence, a slight change to the code was made in order to make it return 1 model for each class in the MNIST dataset resulting in 10 different models. The alteration to the code was to simply have an input vector of 10 1's in the forward, backward and sample functions of this model, and then simply train each model on the respective class that the model should classify.

The Autoencoder is used to reduce the dimensions of a given input and has a similar structure to the discriminative classifier previously described in the report as can be seen in [14], however the Autoencoder returns an embedding for the image which in this case is a list of 20 values corresponding to the given image. These values can then be passed through a decoder given by the Autoencoder in order to recreate the image. The reasoning behind using an Autoencoder for the RealNVP, is that it helps the RealNVP learn the structure of the image.

Instead of using stochastic gradient descent to optimize the Autoencoder the so called Adam algorithm was applied instead [15]. In order to compute the loss for the Autoencoder binary cross entropy loss was applied between the decoding of the embedding of the given image and the image itself. Binary cross entropy loss is generally used to measure the error of a reconstruction as is mentioned in [16], this is quite usefull when training an Autoencoder as the objective for an Autoencoder is to ideally reconstruct an image.

To train the RealNVP models, first the trained Autoencoder was applied to change the MNIST dataset to a dataset of embeddings. Then this dataset was split into 10 different datasets where each one of these new datasets only had 1 class (dataset 0 only had 0's in it etc). Then these datasets were sent into 10 different RealNVP models in order to train the classifier. The optimizer used for the training here was also the Adam algorithm and the loss function was the one described in equation 8.

The Autoencoder was trained for 10 epochs, the learning rate for the Autoencoders optimiser was set to 10^{-3} and the weight decay was set to 10^{-5} . In case of the RealNVP, each model was trained for 20 epochs, the learning weight for this optimiser was set to 10^{-4} and the weight decay was set to 10^{-5} .

C. Training on Gaussian noise

The classifiers were also trained on Gaussian noise in order to check the difference in performance when this training method was applied. For the discriminative classifier the training was similar, however the training set was split in 2 equally sized subsets. One of the subsets included images with no noise applied on it, the other subset included only images with Gaussian noise applied to them, the chosen standard deviation for this noise was 0.5 and the mean was 0. The classifier was then trained on this training set, using the same amount of epochs and the same learning rate.

In order to train the RealNVP on Gaussian noise the training set had to first be split into 10 different subsets. Each subset included only images of a certain class (for example subset 4 only had images illustrating a 4), this was done as otherwise some RealNVP models might not get any noisy images and some of the RealNVP models might only get noisy images, which would create an unwanted bias in the classifier. After the training set had been split into these 10 subsets, Gaussian noise with a standard deviation of 0.5 and mean of 0 was applied on half of each subset. After which the subsets were merged into 1 training set. Once this was completed the Autoencoder was trained using this training, then the RealNVP models were trained using the embeddings from the Autoencoder trained on this training set. This training was conducted with the same amount of epochs, learning rate and weight decay for both the Autoencoder and the RealNVP models.

D. Fast gradient sign method

The fast gradient sign method to generate adversarial noise was simply implemented by applying a function to each image. This function was given the image, the value ϵ (which was discussed in equation 11) and the gradient of the data with respect to the model parameters. The function then simply applied equation 11 in order to generate the noise, then the perturbed image was clamped between 0 and 1 in order to confirm that all of the values in the image was between 0 and 1.

In order to retrieve the data gradient for the discriminative classifier the method discussed in [17] was used. Here one simply receives the data gradient by inputting the clean image to the discriminative model, then taking a negative log-likelihood loss between the correct class for the image and the output of the discriminative model (a list of values where the index of the highest value corresponds to the correct class). The accuracy plot for this case can be seen as

the orange line in figure 13 and 14.

For the generative classifier there were 2 methods for receiving the data gradient which were applied. Both methods firstly received the gradients from the Autoencoder. The loss function for the Autoencoder was calculated with binary cross entropy loss as mentioned in [16].

- The first method was similar to the method used for the discriminative case, where the log-likelihoods for each model in the classifier was acquired as in equation 8. Then a negative log-likelihood loss was calculated between these log-likelihoods and the correct class for the image.
- For the secondary method the gradients were calculated by simply taking a backwards step on the likelihood function for the RealNVP model corresponding to the class of the image. This methodology also follows the theory for fast gradient sign method, because it uses the class of the image and the parameters of the classifier in order to calculate the gradient.

The resulting accuracy plot for the first method can be seen in figure 14 as the blue line, and the resulting accuracy plot for the second method can be seen in figure 13 as the orange line.

V. TESTING

This section describes how each test was conducted for the different types of noise that were tested. All of the testing was done using a subset of the MNIST dataset, namely a validation set. The validation set was generated by taking the validation set provided by MNIST, which consists of 10000 images. In order to deduce if the classifier made a correct classification, maximum likelihood was applied.

A. Adversarial noise



Fig. 7. Adversarial noise applied on an image with the discriminative classifier with epsilon = 0.2

The adversarial noise was tested on both classifier using both implementations for fast gradient sign method, which was discussed in the previous section. Testing was conducted on the validation set, and ϵ was gradually increased from 0 up



Fig. 8. Adversarial noise applied on an image with the generative classifier with epsilon = 0.2

until and including 0.5, with a step size of 0.05. The plots that were generated for each method for retrieving the data gradient can be seen in figures 14 and 13 respectively.

B. Gaussian noise

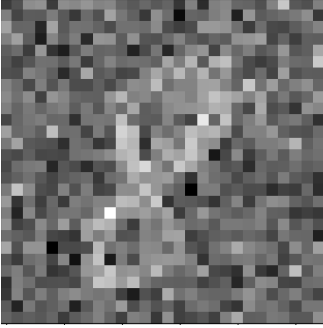


Fig. 9. Gaussian noise: mean = 0, sigma = 0.5

To test the Gaussian noise on the validation set a transformation was applied to each image in the validation set. This transformation added Gaussian noise to every image in the dataset based on the given mean and standard deviation, the mean was always set to 0 for this testing purpose and the standard deviation was increased from 0 to 1 with a step size of 0.1 to produce the plot that can be seen in 15. The test was simultaneously done on both the generative and discriminative classifier to ensure that the validation set was equivalent in both cases. Testing for the classifiers that was trained on Gaussian noise was performed in the exact same manner, the results for these tests can also be seen in 15.

C. Salt and pepper noise

The salt and pepper noise was implemented by picking α pixels at random and colouring half of the pixels white and half of the pixels as black for every image in the validation set. The discriminative and the generative model was then tested on the validation set to test the accuracy of the models after the noise had been applied. The process was then repeated for different values of α and the resulting accuracies was plotted with the corresponding α . The result of this process can be seen in figure 16.



Fig. 10. Salt and pepper noise: alpha = 0.2

VI. RESULTS

A. Clean data results

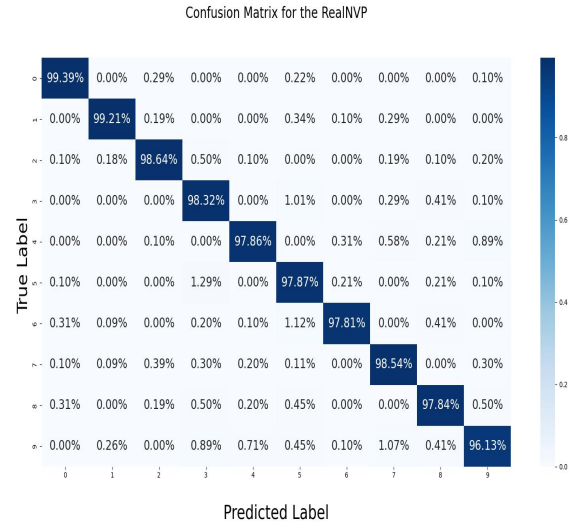


Fig. 11. Confusion matrix for the generative classifier

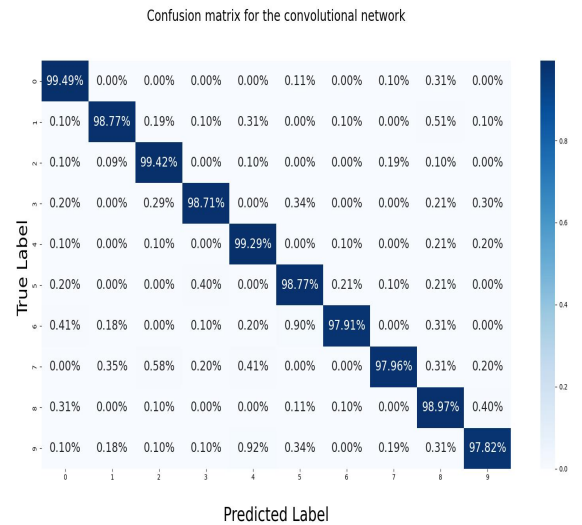


Fig. 12. Confusion matrix for the discriminative classifier

The clean data used to test the models accuracy is the validation set previously discussed in the testing section. These

images had no noise applied to them. The accuracies for the classifiers tested on clean data was 98.71% for the discriminative classifier and 98.18% for the generative classifier. Figures 12 and 11 illustrates the confusion matrices for the generative and discriminative classifiers when tested on clean data. This shows how many correct predictions the classifiers made for each class. The diagonal of the confusion matrix corresponds to a correct prediction, where the first entry in the diagonal shows the accuracy for the classifier on class 0 and the last entry the accuracy for class 9, as is described in [18].

B. Adversarial noise results

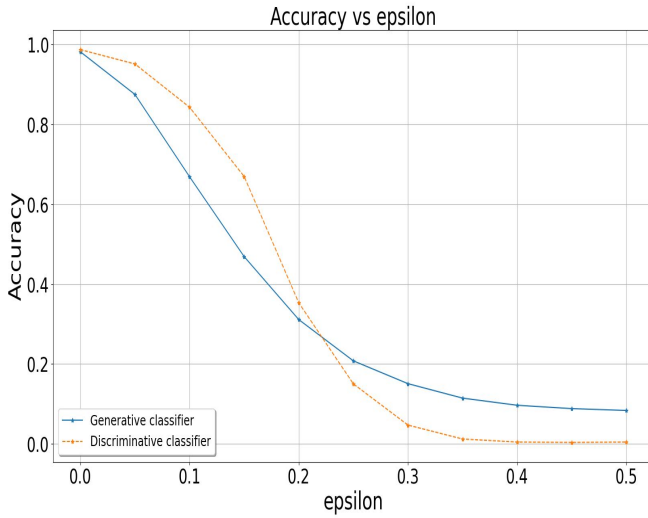


Fig. 13. Accuracy of the classifiers tested on adversarial noise

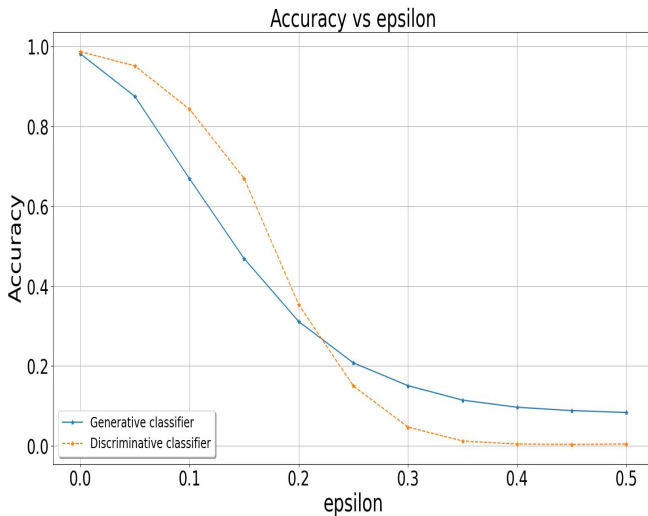


Fig. 14. Accuracy of the classifiers tested on adversarial noise using negative log-likelihood loss

From figure 13 one can deduce that the accuracy for the generative classifier does not converge towards 0 as rapidly as the discriminative classifier does, when the data given to the classifier is adversarial noise. However, the discriminative classifier seems to outperform the generative classifier for small

values of ϵ . Figure 14 seems to show a similar performance to that of 13, even though different methods was applied to retrieve the data gradient.

C. Gaussian noise results

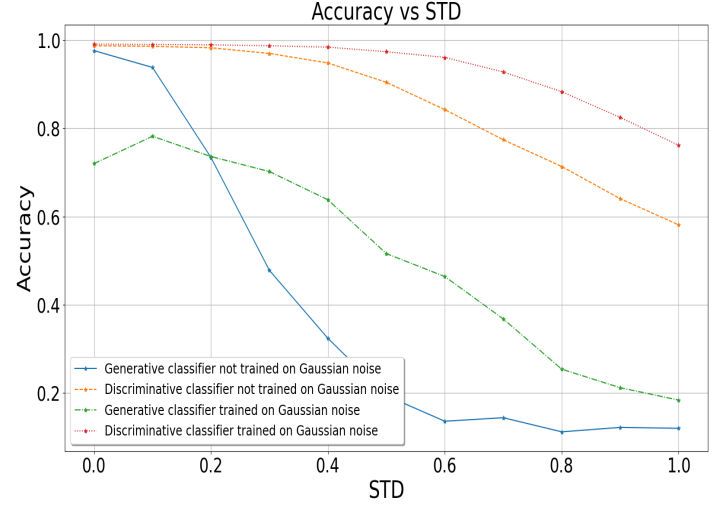


Fig. 15. Accuracy of the classifiers tested on Gaussian noise

Figure 15 shows the difference in accuracy when Gaussian noise has been applied to the data. Here one can deduce that the discriminative classifier outperforms the generative classifier by quite a large margin.

In figure 15, one can also see both of the classifier that has been trained and validated on data with Gaussian noise, the plot shows the accuracy on the validation set. This test gives a rather undesirable result for the generative classifier, as the accuracy for low standard deviation values is around 80%, however one can also note that the classifier has indeed become more robust as it does not seem to converge towards an accuracy of 10% as rapidly, and in the given figure it does not even reach such a value. However, for the discriminative classifier one can identify that the classifier has only gotten more robust when using this training method.

D. Salt and pepper noise results

Figure 16 shows that both classifiers converge towards an accuracy of 10%. This is expected as the classifiers has no way to discern the images once all of the pixels has been altered by the salt and pepper noise, hence the classifiers are simply giving random guesses once the percentage of pixels changed are at 100%. However, the generative classifier seems to converge quicker towards a 10% accuracy. This is also expected as the generative classifier is more biased towards images that are within the set that it has been trained on.

VII. DISCUSSION

As the accuracy for the clean data is so high (98.18% for the generative classifier and 98.71% for the discriminative

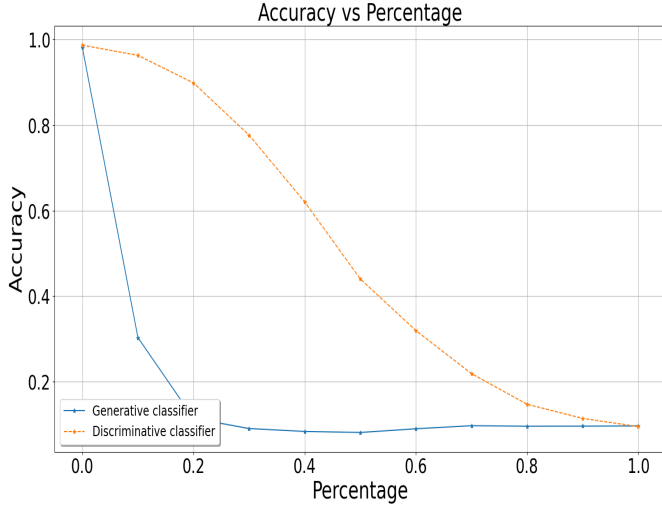


Fig. 16. Accuracy of the discriminative classifier on salt and pepper noise

classifier), it can be deduced that there is no bias towards which images in the MNIST dataset are being tested. This is very important as otherwise the test results for noisy data may be incorrect, as the validation set could include images which one of the classifiers consistently predict incorrectly when the image is clean. Hence, this leads to a more accurate result when analyzing noise. When analyzing the confusion matrices from figures 12 and 11, it can be identified that there is no such bias towards any class of images.

From the results for the adversarial noise, one can deduce that the coupling flow RealNVP performs better than the convolutional network when tested on adversarial noise, atleast this seems to be the case when ϵ increases. As ϵ increases the RealNVP classifier seems to converge towards an accuracy of 10%, however the convolutional network seems to converge towards an accuracy of 0% as can be seen in figures 14 and 13. However, the only objective conclusion that can be drawn from the results is that the RealNVP classifier performs worse than the convolutional network for $\epsilon \in [0, 0.2]$ and the RealNVP classifier outperforms the convolutional network when $\epsilon \in [0.25, 0.5]$, this holds true for both methodologies of achieving the datagradient for the fast gradient sign method. In these results one can also note that the two methods that were proposed for generating the data gradient seem to be giving almost equivalent results, as is seen in figure 14 and 13. The reasoning behind this is unknown to the authors. However, the reason may be due to the fact that the fast gradient sign method only uses the sign of the gradient to compute the noise that shall be applied to the image. Hence, the case may be that the computed gradients have equivalent directions, however they might not have equivalent values.

It should also be noted that retrieving the data gradient via the use of cross entropy between the targeted class value and the list of likelihood values given by the classifier, could

also be done by applying a LogSoftmax to the likelihood values before hand. This would most likely generate slightly different results as the LogSoftmax would cause the difference between values in the list of likelihood values to be greater. The LogSoftmax is simply the logarithm of the Softmax function which can be seen in equation 16 as is described in [19].

$$\sigma(z)_i = \log\left(\frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}\right) \quad (16)$$

One could also consider not taking the gradient of the Autoencoder into account when perturbing the image with the fast gradient sign method. Such a methodology would only take into account the RealNVP model, and a different result should be given.

In the case for the Gaussian noise tests one can conclude that both models seem to perform better on Gaussian noise once they have been trained with Gaussian noise, as is seen in figure 15. However, training with Gaussian noise seems to make the generative classifier perform worse when no noise is applied to the validation set, however it performs a considerable amount better when noise is applied to the dataset. Hence, one can conclude that for discriminative classifiers it can be a good idea to apply Gaussian noise to some parts of the training dataset as it only makes it more robust. However, for the generative case one could instead consider using a denoising Autoencoder (DAE) instead of a regular Autoencoder as is discussed in [20], instead of attempting to train both the Autoencoder and the generative classifier on Gaussian noise. From this one can deduce that if the images were to be at high risk to be perturbed with Gaussian noise, either a discriminative classifier should be used to classify said images or a generative classifier with the addition of a DAE.

When analysing the results of the salt and pepper noise it can be seen that the discriminative classifier performs better than the generative classifier for all different noise levels except when the data is clean or when all of the pixels are noisy. This leads to the conclusion that the discriminative classifier is more robust in applications where there is a high risk of salt and pepper noise, for instance when an image is transmitted over a noisy digital link as was stated in the theory section of the report.

VIII. FUTURE WORK

An interesting continuation of this project would be an attempt at constructing a robust classifier which performs well on adversarial noise. One could for example examine how well the classifiers perform once they have been trained on some adversarial noise. Another interesting continuation could be the construction of a classifier which is resistant to the other types of noise tested in this project (Gaussian, salt and pepper noise), or further analysis could be done to see which generative classifier give the best performance for adversarial noise testing. For example, one could test the differences

between a GMM and normalized flow tested on adversarial noise.

IX. CONCLUSION

Conclusively it can be deduced that the effect of adversarial noise is significant on both image classifiers tested in this project. Neither classifier performed particularly well, however the generative classifier that was tested seemed to not converge towards an accuracy of 0%. From the results one can also conclude that the effect of both Gaussian and salt and pepper noise are greater on generative classifiers than that of discriminative classifiers. Hence it can be deduced that the discriminative classifier is more robust on both Gaussian and salt and pepper noise. Training on Gaussian noise seems to not be a particularly good idea for the case of generative classifiers as it will induce a worse performance for clean data. However, in the case of the discriminative classifier this training only made the classifier more robust.

ACKNOWLEDGMENT

The authors would like to thank supervisors Anubhab Ghosh and Saikat Chatterjee for their help in this project. Anubhab Ghosh provided a great amount of help regarding how to implement the different classifiers, and a lot of sources to help study the theory behind the project. Guidance for the understanding of the theory behind the project was also given directly by Anubhab Ghosh and was appreciated by the project group. It would have been difficult to complete the project without this guidance.

REFERENCES

- [1] Z. You, J. Ye, K. Li, Z. Xu, and P. Wang, "Adversarial noise layer: Regularize neural network by adding noise," in *2019 IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 909–913.
- [2] S. Gu and L. Rigazio, "Towards deep neural network architectures robust to adversarial examples," *arXiv preprint arXiv:1412.5068*, 2014.
- [3] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.
- [4] A. Krogh, "What are artificial neural networks?" *Nature Biotechnology*, vol. 26, pp. 195–197, Feb 2008.
- [5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016, [Online]. Available: <http://www.deeplearningbook.org>.
- [6] S. Dreiseitl and L. Ohno-Machado, "Logistic regression and artificial neural network classification models: a methodology review," *Journal of biomedical informatics*, vol. 35, no. 5-6, pp. 352–359, 2002.
- [7] 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.
- [8] J. Liu, D. Cai, and X. He, "Gaussian mixture model with local consistency," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 24, no. 1, pp. 512–517, Jul. 2010. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/7659>
- [9] I. Kobyzev, S. J. Prince, and M. A. Brubaker, "Normalizing flows: An introduction and review of current methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 11, pp. 3964–3979, 2021.
- [10] F. Luisier, T. Blu, and M. Unser, "Image denoising in mixed poisson–gaussian noise," *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 696–708, 2011.
- [11] A. Bovik, *Handbook of image and video processing*. Austin, Texas: Academic Press, 2000, [Online]. Available: <https://preetikale.files.wordpress.com/2018/07/handbook-of-image-and-video-processing-al-bovik1.pdf>.
- [12] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems* 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: <http://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
- [13] J. P. Simone Scardapane. (2020, Mar) realnvp-demo-pytorch. [Online]. Available: <https://github.com/ispamm/realnvp-demo-pytorch>
- [14] D. Bank, N. Koenigstein, and R. Giryes. (2020) Autoencoders. [Online]. Available: <https://arxiv.org/abs/2003.05991>
- [15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. [Online]. Available: <https://arxiv.org/abs/1412.6980>
- [16] U. Ruby and V. Yendapalli, "Binary cross entropy with deep learning technique for image classification," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, 10 2020.
- [17] N. Inkawhich. (2022, Apr) Adversarial example generation. [Online]. Available: https://pytorch.org/tutorials/beginner/fgsm_tutorial.html
- [18] F. Ariza-Lopez, J. Rodriguez-Avi, and M. Alba-Fernandez, "Complete control of an observed confusion matrix," in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, 2018, pp. 1222–1225.
- [19] P. Blanchard, D. J. Higham, and N. J. Higham. (2019) Accurate computation of the log-sum-exp and softmax functions. [Online]. Available: <https://arxiv.org/abs/1909.03469>
- [20] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, p. 3371–3408, Dec 2010.

Neonatal Sepsis Detection Using Decision Tree Ensemble Methods: Random Forest and XGBoost

Marwan Al-Bardaji and Nahir Danho

Abstract—Neonatal sepsis is a potentially fatal medical condition due to an infection and is attributed to about 200 000 annual deaths globally. With healthcare systems that are facing constant challenges, there exists a potential for introducing machine learning models as a diagnostic tool that can be automatized within existing workflows and would not entail more work for healthcare personnel. The Herlenius Research Team at Karolinska Institutet has collected neonatal sepsis data that has been used for the development of many machine learning models across several papers. However, none have tried to study decision tree ensemble methods. In this paper, random forest and XGBoost models are developed and evaluated in order to assess their feasibility for clinical practice. The data contained 24 features of vital parameters that are easily collected through a patient monitoring system. The validation and evaluation procedure needed special consideration due to the data being grouped based on patient level and being imbalanced. The proposed methods developed in this paper have the potential to be generalized to other similar applications. Finally, using the measure receiver-operating-characteristic area-under-curve (ROC AUC), both models achieved around ROC AUC= 0.84. Such results suggest that the random forest and XGBoost models are potentially feasible for clinical practice. Another gained insight was that both models seemed to perform better with simpler models, suggesting that future work could create a more explainable model.

Sammanfattning—Neonatal sepsis är ett potentiellt dödligt medicinskt tillstånd till följd av en infektion och uppges globalt orsaka 200 000 dödsfall årligen. Med sjukvårdssystem som konstant utsätts för utmaningar existerar det en potential för maskininlärningsmodeller som diagnostiska verktyg automatiserade inom existerande arbetsflöden utan att innebära mer arbete för sjukvårdspersonal. Herlenius forskarteam på Karolinska Institutet har samlat ihop neonatal sepsis data som har använts för att utveckla många maskininlärningsmodeller över flera studier. Emellertid har ingen prövat att undersöka beslutsträds ensemble metoder. Syftet med denna studie är att utveckla och utvärdera random forest och XGBoost modeller för att bedöma deras möjligheter i klinisk praxis. Datan innehåller 24 attribut av vitalparameterar som enkelt samlas in genom patientövervakningssystem. Förfarandet för validering och utvärdering krävde särskild hänsyn med tanke på att datan var grupperad på patientnivå och var obalanserad. Den föreslagna metoden har potential att generaliseras till andra liknande tillämpningar. Slutligen, genom att använda receiver-operating-characteristic area-under-curve (ROC AUC) måttet kunde vi uppvisa att båda modellerna presterade med ett resultat på ROC AUC= 0.84. Sådana resultat föreslår att både random forest och XGBoost modellerna kan potentiellt användas i klinisk praxis. En annan insikt var att båda modellerna verkade prestera bättre med enklare modeller vilket föreslår att framtida arbete skulle kunna vara att skapa en mer förklarlig maskininlärningsmodell.

Index Terms—Machine Learning, Sepsis, Neonatal Sepsis, Random Forest, XGBoost, Imbalanced Data, Binary Classification, Cross-Validation, Hyperparameter Tuning.

Supervisors: Antoine Honoré

TRITA number: TRITA-EECS-EX-2022:175

I. INTRODUCTION

The goal of the healthcare system is the maintenance and improvement of the health of a population. An essential step in that work is the diagnosis and detection of diseases, where physicians typically work using a combination of their own experience and medical guidelines using years of research [1]. This project focuses on neonatal sepsis, a possibly fatal medical condition due to an infection [2]. Globally sepsis is attributed to about 200 000 annual deaths [3]. There exist several scoring systems for the detection and prognosis estimation of sepsis built on international consensus; however, none is perfect [2], [4]. Meanwhile, healthcare systems are constantly facing issues such as rising costs [5], staffing shortages [6], and aging populations with ever-increasing healthcare needs. This could lead to situations where guidelines cannot be followed perfectly, thus endangering patient safety [1]. Any method that could increase diagnostic performance without inferring increased effort by healthcare personnel would be desirable. Machine learning models have been on the rise since the 20th century, and adoption is continuously increasing with more powerful computers and the improved ability to collect large datasets. There exist many machine learning algorithms whose goal is to classify data and could therefore have the potential as a diagnostic tool in healthcare that can be automatized within existing workflows [7].

A. Problem Formulation

This project will use neonatal sepsis data provided by the Herlenius Research team at Karolinska Institutet. Earlier work for the detection of sepsis using machine learning models has been conducted using the same data. Examples of previous models that have been used are Markov models, logistic regression, naïve Bayes, multi-layer perceptrons, Gaussian mixture models, and normalizing flows [8], [9]. Many of the models achieved promising results. Nevertheless, no work has studied the performance of models that use decision tree ensemble methods.

B. Project Goal and Scope

The goal of the project is to study the feasibility of using decision tree ensemble methods to detect neonatal sepsis.

Specifically, the performance of random forest and XGBoost models will be evaluated and compared. Moreover, insightful learnings from the development of the models will be collected and discussed. The project is limited to a proof of concept and does not create a model ready for real-life implementation.

II. BACKGROUND

A. Machine Learning

Machine Learning (ML) is a subset of artificial intelligence that deals with learning in the sense that the algorithm's performance on future tasks can be improved by making observations of the world [7]. Machine learning algorithms are generally divided into three categories (1) supervised learning, (2) unsupervised learning, and (3) reinforcement learning. Supervised learning is predictive, where the algorithm maps an input to an output. Hidden training data is provided for which the mapping is known beforehand, also known as labeled data. A trained model can later be used to predict the output of future inputs with unknown outputs. Unsupervised learning analyzes unstructured data and tries to find a pattern on its own; here, there exists only input and no output. On the other hand, reinforcement learning is an algorithm that reacts to the environment rewarding desired behaviors and punishing undesired ones [7]. There exists a myriad of algorithms under each category; moreover, there also exist algorithms that do not fall under one category, some of which are semi-supervised learning. This project will deal with a classification problem within supervised learning.

B. Medical Background

Sepsis is a possibly fatal medical condition which, in layman's terms, often somewhat incorrectly is called blood poisoning, thus not revealing the entire truth. According to international consensus [2], sepsis should be defined as "life-threatening organ dysfunction caused by a dysregulated host response to infection. Left untreated, sepsis can turn into sepsis shock. sepsis shock is defined as [2] "a subset of sepsis in which underlying circulatory and cellular metabolism abnormalities are profound enough to increase mortality substantially." Many survivors develop permanent neurologic impairment [10]. Sepsis does not have obvious symptoms, especially in the early course of the condition. To identify sepsis in clinical practice, international consensus recommends using the Sequential Organ Failure Assessment (SOFA) score or the quick SOFA (qSOFA) score [4], [11], [12]. These scoring systems use a combination of vital parameters, such as partial pressure of oxygen in the blood and blood pressure; blood tests, such as platelet count and bilirubin; and neurological status [11]. Moreover, after a sepsis suspicion has arisen, there exist international guidelines created by the "Surviving Sepsis Campaign" for the most appropriate tests and treatments to continue with [12]. Suspected sepsis can be confirmed through microbiologic blood cultures; the culture needs to be obtained before any antibiotic treatments. In principle, there exist two types of treatment (i) antimicrobial treatment, which usually starts with empiric broad-spectrum antibiotics, and (ii) organ-supportive such as fluid therapy and vasoactive medication

[12]. An inherent limitation of using the SOFA scoring system is that doctors and nurses conduct additional tasks such as taking blood tests that do not necessarily need to happen for all patients [11].

This project will study neonatal sepsis, meaning sepsis in the first weeks of an infant's life. In neonates, sepsis is difficult to diagnose clinically since they may be asymptomatic until organ dysfunction is prominent [13]. There exists an adapted version of the SOFA score called the nSOFA score; however, this score still requires invasive testing such as blood tests which both are costly and take time compared to collecting vital parameters. Moreover, the nSOFA scoring is not widely adopted [14]. A study by Fairchild suggests that there exists potential with using heart rate variability, heart rate characteristics, and other vital signs in the detection of Neonatal Sepsis. These measurements are non-invasive and can easily be collected through a monitoring machine [10]. Current challenges in the treatment of neonatal sepsis include late detection, overuse of antibiotics, and difficulties with invasive testing [10].

C. Classification Problems

A classification problem within supervised machine learning refers to a predictive modeling problem where the aim is to predict an output given an input. The following definition of a classification problem will assume a one-dimensional output, also called the label. Each data input will be provided as a feature vector \vec{x}_i with its corresponding output y_i . Since the input is in vector form, it may include multiple data points corresponding to several features.

To train the machine learning model, a dataset \mathcal{D} with n examples and m features is provided.

$$\mathcal{D} = \{(\vec{x}_i, y_i)\} \quad (|\mathcal{D}| = n, \vec{x}_i \in \mathbb{R}, y_i \in \mathbb{R}) \quad (1)$$

(Binary classification is the case when $y_i \in \mathbb{B} = \{0, 1\}$).

A prediction function ϕ gives the prediction.

$$\hat{y}_i = \phi(\vec{x}_i) \quad (2)$$

The behavior of ϕ depends on internal model parameters that are unique for each machine learning implementation. The procedure of finding the internal parameters is what training a model means [7]. The internal parameters of a model are fit by minimizing the objective function \mathcal{L} which typically is of the following form.

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \Omega(\theta) \quad (3)$$

Here l is a differentiable convex loss function that measures the difference between y_i and \hat{y}_i . The second term Ω is a function of the internal parameters which penalize complex models. There exist several loss and penalty functions depending on the implementation that shares the concept of distance and model complexity, respectively. [7]

If an unseen data point \vec{x}^{unseen} is provided without a label then there exists no way to know whether the prediction $\hat{y}^{\text{unseen}} = \phi(\vec{x}^{\text{unseen}})$ is correct. However, if a theoretical label y^{real} is allowed to exist then the goal of any classification

algorithm is that $\hat{y} = y^{\text{real}}$. To imitate this procedure, the original dataset \mathcal{D} is commonly divided into a test set and a training set.

The training set is used for fitting the internal parameters of the selected model, while the test set is used to evaluate the performance of the model. Sometimes the original dataset is divided into yet another set called the validation set to optimize external parameters, also called hyperparameters, on unseen data before evaluating the results on the test set. External parameters affect a model's behavior and do not change depending on the training data [7].

D. Decision Trees

A decision tree is a representation of a function that maps a feature vector to a single output value which is called the decision [7]. In essence, it is similar to a flowchart of questions that are commonly used within the healthcare system [15]. The decision tree starts with a root node. Each node may, in turn, split into several other nodes. A node that does not split into other nodes is called a leaf and contains the final decision. At each node, a question regarding the question is asked. In implementations using numerical data the question are comparisons of the types $<$, \leq , $=$, \geq , $>$.

The prediction functions ϕ works by using an algorithm that finds the feature and the question to ask that provide the highest "importance." Importance is measured using information gain, which is defined in terms of entropy. These quantities are fundamental in information theory. To train the decision tree, all the feature vectors in the training data set to go through the entire decision tree, and the number of samples and their class are calculated at the leaf nodes [7], which means that each node sorts the incoming data into smaller sets.

Entropy is a measure of uncertainty of a random variable; the more information, the less entropy. In general, the entropy of a random variable V with values v_k having the probability $P(v_k)$ is defined as [7].

$$\text{Entropy} = H(V) = - \sum_k P(v_k) \log_2 P(v_k) \quad (4)$$

Entropy is measured in bits and corresponds to the expected number of 50/50 guesses it would require to narrow down to a specific value v_k . For example, a fair coin flip has an entropy of $H(\text{Fair coinflip}) = -2 \cdot (0.5 \log_2 0.5) = 1$ and a fair six sided die has an entropy of $H(\text{Fair six-sided die}) = -6 \cdot (1/6 \log 1/6) \approx 2.6$ [7].

The information gain at a node is calculated as the expected reduction in entropy by sorting the incoming data. The information gain on attribute A and data S is defined as [7].

$$\text{Information gain} = H(S) - \sum_{i \in v} \frac{|S_i|}{|S|} H(S_v) \quad (5)$$

where S is the incoming data to the node, v is a set of mutually exclusive questions, and S_i is the sorted version of S after asking a question i . S is considered to be a random variable with values and probabilities according to the distribution in the data [7].

The maximization of the information is thus the objective

function of the decision tree. However, in order to limit the complexity of the model and reduce bias, it is possible to penalize complex models with a regularization parameter Ω [7].

E. Random Forests

Random forests are an ensemble of decision trees that uses bootstrap aggregating to reduce variance in a noisy dataset by training multiple different trees. A majority vote of all trees then decides the output [16].

A random forest classifier R is defined as

$$R = \text{Majority vote of } \{h(\vec{x}, \Theta_k), k = 1, \dots\} \quad (6)$$

Where h is a decision tree, and Θ_k is an independent random vector that the decision tree k uses in the construction of the tree [16].

The primary source of randomness in a random forest is feature subsampling, where a random number of features are selected for each tree. This reduces the bias by increasing the probability that the trees are uncorrelated. The objective function and penalization of complex trees are analogous to the decision tree [16].

F. XGBoost

XGBoost, an implementation of gradient boosted decision trees, has shown state-of-the-art results in many machine learning challenges. XGBoost is an ensemble method of regression trees that uses boosting, which aims to improve performance by creating a strong classifier from many weak classifiers. Regression trees use the same concepts as a decision tree but have a continuous target variable. Moreover, XGBoost is designed with system performance in mind and is easily scalable [17].

1. Objective Function

For a given dataset with n examples and m features

$$\mathcal{D} = \{(\vec{x}_i, y_i)\} \quad (|\mathcal{D}| = n, \vec{x}_i \in \mathbb{R}, y_i \in \mathbb{R}) \quad (7)$$

A tree ensemble method uses K additive functions to predict the output according to

$$\hat{y} = \phi(\vec{x}_i) = \sum_{k=1}^K f_k(\vec{x}_i), \quad f_k \in \mathcal{F} \quad (8)$$

where $\mathcal{F} = \{f(\vec{x}) = w_{q(\vec{x})}\} \quad (q: \mathbb{R}^m \rightarrow T, w \in \mathbb{R}^T)$ is the space of regression trees. T is the number of leaves in each tree and q represents the structure of each tree that maps an input to the leaves. Each tree structure q and leaf weights w correspond to a specific f_k .

To learn the weights w used in the model, the following regularized objective is minimized [17].

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \quad (9)$$

$$\text{where } \Omega(f) = \gamma T + \frac{1}{2} \lambda \|w\|^2 \quad (10)$$

Here l is a differentiable convex loss function, \hat{y}_i is the prediction and y_i is the target. The second term penalizes

a complex model regarding the size of the weights and the number of leaves [17].

2. Gradient Tree Boosting

Since the equation

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \quad (11)$$

contains functions as parameters it cannot be optimized using traditional optimization methods in Euclidean space and has to train in an additive manner. The prediction of instance i at iteration t is defined as $\hat{y}_i^{(t)}$. Then $\hat{y}_i^{(t)}$ is calculated by adding $f_t(\vec{x}_i)$ to $\hat{y}_i^{(t-1)}$ [17]. Therefore the objective to minimize turns into

$$\mathcal{L}^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(\vec{x}_i)) + \Omega(f_t) \quad (12)$$

The greedy approach is to add the f_t that improves the model the most. By using a second-order approximation of the objective, it turns into

$$\mathcal{L}^{(t)} \simeq \sum_{i=1}^n [l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(\vec{x}_i) + \frac{1}{2} h_i f_t^2(\vec{x}_i)] + \Omega(f_t) \quad (13)$$

$$\text{where } g_i = \frac{\partial l(\hat{y}_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}} \text{ and } h_i = \frac{\partial^2 l(\hat{y}_i, \hat{y}_i^{(t-1)})}{\partial (\hat{y}_i^{(t-1)})^2} \quad (14)$$

A simplified objective function $\tilde{\mathcal{L}}^{(t)}$ at step t is obtained by removing the constant terms

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^n [g_i f_t(\vec{x}_i) + \frac{1}{2} h_i f_t^2(\vec{x}_i)] + \Omega(f_t) \quad (15)$$

Define $I_j = \{i | q(\vec{x}_i) = j\}$ as the instance set of leaf j meaning all the instances that correspond to the leaf. The equation can be rewritten by expanding Ω

$$\begin{aligned} \tilde{\mathcal{L}}^{(t)} &= \sum_{i=1}^n [g_i f_t(\vec{x}_i) + \frac{1}{2} h_i f_t^2(\vec{x}_i)] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \\ &= \sum_{j=1}^T \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] + \gamma T \end{aligned} \quad (16)$$

For a fixed structure $q(\vec{x})$ the optimal weight w_j^* of leaf j is calculated by

$$w_j^* = - \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \quad (17)$$

The corresponding optimal value is given by

$$\tilde{\mathcal{L}}^{(t)} = - \frac{1}{2} \sum_{j=1}^T \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T \quad (18)$$

Equation 15 can be used as a scoring function to measure the quality of the tree structure q . Usually, it is impossible to enumerate all the possible tree structures q . A greedy algorithm is developed by starting from a single leaf and iteratively adding. Assume that I_R and I_L are instance

sets of the right and left nodes after a split. Letting $I = I_R \cup I_L$ then the loss reduction after the split is given by [17]

$$\mathcal{L}_{\text{split}} = \frac{1}{2} \left[\frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} + \frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda} - \gamma \right] \quad (19)$$

Equation 16 is the formula that is used in practice in XGBoost [17].

3. Split Finding Algorithms Using equations 14 and 15, two algorithms for split finding can be written according to the following pseudocode [17].

Algorithm 1 Exact Greedy Algorithm for Split Finding

Input: I , instance set of current node

Input: d feature dimension

$gain \leftarrow 0$

$G \leftarrow \sum_{i \in I} g_i, H \leftarrow \sum_{i \in I} h_i$

for $k=1$ **to** m **do**

$G_L \leftarrow 0, H_L \leftarrow 0$

for j **om** sorted(I , by \vec{x}_{jk}) **do**

$G_L \leftarrow G_L + g_j, H_L \leftarrow H_L + h_j$

$G_R \leftarrow G - G_L, H_R \leftarrow H - H_L$

end for

end for

Output: Split with maximum score

Algorithm 2 Approximate Algorithm for Split Finding

for $k=1$ **to** m **do**

Propose $S_k = \{s_{k1}, s_{k2}, s_{k3}, \dots, s_{kl}\}$ by percentiles on feature k

Proposal can be done per tree (global), or per split(local).

end for

for $k=1$ **to** m **do**

$G_{kv} \leftarrow \sum_{j \in \{j | s_{k,v} \geq \vec{x}_{jk} > s_{k,v-1}\}} g_j$

$H_{kv} \leftarrow \sum_{j \in \{j | s_{k,v} \geq \vec{x}_{jk} > s_{k,v-1}\}} h_j$

end for

Follow same step as in previous section to find maximum score only among proposed splits.

III. METHODS

A. Selection of Machine Learning Models

A binary classifier was to be created that could distinguish sepsis using patient data. The random forest and XGBoost models were selected to study their potential for this endeavor using the implementations of the sci-kit learn and XGBoost libraries in python, respectively [18], [19].

B. Study population

Data was acquired from the Herlenius Research team at Karolinska Institutet that are currently conducting research regarding neonatal healthcare. The population consisted of

very low birth weight infants (< 1500 g) hospitalized in the Neonatal Intensive Care Unit (NICU) at Karolinska University Hospital, Stockholm, Sweden [9].

C. Data Description

Time-series data were collected from all patients was collected from a high-frequency Phillips IntelliVue MX800 Patient Monitor [9]. The data initially contained monitor data sampled at 1 Hz. The times-data was split into windows of 70 minutes where 19 different features were extracted. If a patient received a sepsis diagnosis at a specific time instance, all the windows 24 hours before the time of the diagnosis was categorized as sepsis-like. Sepsis-like windows were set to the value 1, and non-sepsis-like time windows were set to the value 0, and these values corresponded to the target y_i in our binary classification problem. Moreover, five more features were also provided that included parameters regularly updated in the medical history. In total, the data consisted of 24 features. Table I provides a list of the target label and all the features that were provided in the data.

Data is provided in a tabular manner where each data row corresponds to a specific time window. The data initially contained 134668 data rows from 118 different patients. A total of 10 patients experienced sepsis during their hospitalization together, totaling 556 rows included sepsis-like characteristics, thus having a target_y = 1.

It can be noticed that many features seem to contain large negative values down to -99999 ; also there appears to exist missing data for the `feats_cirk_vikt` feature. A more detailed description of the data, including the median and percentiles, can be found in the appendix of the project.

D. Preprocessing

The data in its unmodified state is not universally usable due to several reasons. The following issues have been identified and need to be dealt with. When discussing the following naming convention is used in this project: the feature vector corresponds to a row, and one specific value in the feature vector is called a data point.

- **Erroneous data points** - Some data points have been identified where the value of some features are exactly -99999 or -9999 and are far outliers relative to the rest of the data. The reason for this data is related to errors during the patient monitoring data collection.
- **Categorical data** - The unmodified data includes the `feats_group_uid` feature, which is categorical. Many machine learning algorithms do not work with categorical data [7].
- **Large variations in the data** - There exist large variations in the data which may affect the weights of the internal parameters in distance-based models [7].
- **Missing data** - One of the features has missing data that needs to be dealt with. The unmodified data contains 6595 missing data points in only the `feats_cirk_vikt` feature. If all the rows containing a missing data point were removed, it would result in an additional 158280 data points lost. This is equal to almost 5% of all rows.

With a background in solving the aforementioned issues, the following data preprocessing steps are taken. These steps aim to prepare the data for the random forest and XGBoost algorithms but also prepare for other eventual algorithms that would want to be tested in the future, including distance-based machine learning algorithms.

1. **Label Encoding** - The 'group_uid' for the entire dataset is encoded using a Label Encoder, creating a 1-to-1 mapping from the String domain to the integer domain. This is to facilitate future slicing and grouping of the dataset.
2. **Removal of Erroneous data points** - Some of the data points from features that initially were collected from the patient monitor was exactly equal to -99999 and -9999 . Since the probability of achieving an exact integer in any continuous distribution is zero, there exists no risk of real data being removed.
3. **Scaling** - This part is not necessary for random forests and XGBoost. However, regarding distance-based algorithms, the internal parameters are affected by the values of the datapoint. Having a large variance in the sizes of the data points makes it more difficult to fit appropriate internal parameters [7]. All the data except 'target_y' and 'group_uid' is scaled using a standard scaling that removes each data point's mean within a feature and scales to unit variance.
4. **Missing Data** - The remaining missing data points are assigned values based on the neighboring data points. More specifically, a KNN imputer is used, which stands for K nearest neighbor imputer. Initial testing resulted in the selection of $K = 5$ due to not significantly changing the mean and the standard deviation of the `feats_cirk_vikt` feature. In other words, the algorithm looks at the five nearest neighbors. The specifics of how the algorithm works are explored in further detail in the appendix to the project.

Table I contains a description of the data after removing erroneous data points. A more detailed description describes the data after each step also, including the median and percentiles, can be found in the appendix to the project.

E. Cross-Validation for Highly Imbalanced Grouped Time Series Data

Data is needed both to train the internal parameters and to test the performance of the model data. Learning the internal parameters of a prediction function and testing on the same data results in a methodological mistake. The model could simply repeat the labels that it has seen without learning any patterns [7]. A trivial method to carry out such a split would be to randomly select a specific percentage in the training set and the rest to be in the testing set. Typical values are to select about 70% to 80% in the training set [20]. However, when trying different hyperparameters of the models, there may exist situations where part model finds patterns in the training set that does not exist in the testing set. This situation is called overfitting, where the patterns found do not display the entire truth of the data. Moreover, indirect knowledge about

TABLE I
FEATURES THAT WERE PROVIDED AND A DESCRIPTION OF THE DATA AFTER ERRONEOUS DATA POINTS HAVE BEEN REMOVED

Feature Name	Description	Count	Mean	Std	Min	Max
feats_btb_mean	The mean value of the beat to beat interval over a time window.	95849	0.000203	0.00104	-0.0284	0.0239
feats_rf_mean	The mean value of the respiratory frequency over a time window.	95849	50.2	11.5	8.01	97.0
feats_spo2_mean	The mean value of the oxygen saturation over a time window.	95849	93.5	3.47	35.6	100.0
feats_btb_std	The standard deviation value of the beat to beat interval over a time window.	95849	0.0201	0.0144	0.000556	0.241
feats_rf_std	The mean value of the respiratory frequency over a time window.	95849	13.5	4.30	0.0	38.1
feats_spo2_std	The mean value of the oxygen saturation over a time window.	95849	4.31	2.73	0.0	27.8
feats_btb_maximum	The maximum value of the beat to beat interval over a time window.	95849	0.146	0.149	0.0016	1.64
feats_rf_maximum	The maximum value of the respiratory frequency over a time window.	95849	89.7	16.52	15.3	163
feats_spo2_maximum	The maximum value of the oxygen saturation over a time window.	95849	99.7	0.911	79.9	100.0
feats_btb_minimum	The minimum value of the beat to beat interval over a time window.	95849	-0.0430	0.0265	-0.808	-0.00140
feats_rf_minimum	The minimum value of the respiratory frequency over a time window.	95849	20.1	6.85	0.394	55.3
feats_spo2_minimum	The minimum value of the oxygen saturation over a time window.	95849	74.5	14.7	0.0333	100.0
feats_btb_skew	The sample skewness of the beat to beat interval over a time window.	95849	2.18	2.93	-5.24	14.8
feats_rf_skew	The sample skewness of the respiratory frequency over a time window.	95849	0.151	0.626	-17.8	12.7
feats_spo2_skew	The sample skewness of the oxygen saturation over a time window.	95849	-1.34	1.19	-23.2	2.51
feats_btb_kurtosis	The kurtosis of the beat to beat interval over a time window. A measure of the "tailedness" of the sampling points.	95849	20.4	31.8	-1.33	240
feats_rf_kurtosis	The kurtosis of the respiratory frequency over a time window. A measure of the "tailedness" of the sampling points.	95849	0.0564	4.33	-3.0	341
feats_spo2_kurtosis	The kurtosis of the oxygen saturation over a time window. A measure of the "tailedness" of the sampling points.	95849	4.01	10.3	-3.0	592
feats_btb_sampAs	The sample asymmetry of the beat to beat interval over a time window. Assessing the asymmetry of the sampling points [10].	95849	3.64	5.72	0.0652	276
feats_btb_sampEn	The sample entropy of the beat to beat interval over a time window. Assessing the complexity and thus possible deterioration [10].	95849	0.410	0.150	0.0124	1.01
feats_cirk_vikt	The weight of the patient during the time window. (Not sampled through the monitor)	91550	1.28	0.470	0.497	3.72
feats_bw	The birth weight of the patient. (Not sampled through the monitor)	95849	839	258	400	150
feats_sex	Categorical data with values of biological sex at birth. Either 1 or 2 for males and females respectively. (Not sampled through the monitor)	95849	1.56	0.496	1.0	2.0
feats_pnage_days	Patient age in days since birth. Negative days are possible due to inconsistencies with registration of exact birth time. (Not sampled through the monitor)	95849	31.3	22.2	0.0250	13
feats_group_uid	An anonymized personalized string for each patient. (Not sampled through the monitor)					
target_y	1 for sepsis-like behavior over a time window, 0 for non-sepsis like.					

the testing set through iterative evaluation of the model may "leak" into the choice of hyperparameters, and in such an instance, the testing set does not show an unbiased picture of the model's performance. A possible option to overcome this problem is to partition the original dataset into three parts where a validation set is added. This data is not touched until the final valuation of the model after the hyperparameters have been found. Nevertheless, this partitioning drastically reduced the number of samples available for training the model [21]. Since the available data was heavily imbalanced, with a small proportion of the data having positive labels, a partition of the dataset into three parts would result in partitions with only about 100 to 200 sepsis-like rows.

A solution to reduce overfitting without using a validation set is to use cross-validation. The principle idea is that the model is trained several times for a specific hyperparameter choice with different choices of test sets each time. Every instance of model training is called a split. If the model would overfit one of the splits, then an eventual loss of generalization due to overfitting would affect the performance of the other splits. There exist several types of methods of cross-validation, and the choice of the cross validator depends on the data. The data used in this report had the following characteristics that need to be taken into account. (i) The data is heavily imbalanced, with less than one percent of the data points having a positive label. If a split is selected entirely at random, there exists a probability that no positive labels are included in either the training or the testing set of that split. (ii) The data included time-series data grouped on a patient basis. It is salient for any cross-validation to take into not having data rows from the same patient in both the training set and test set of the split. This would result in a methodological error since the model possibly would train on past data and test on future data from the same patient finding patient-specific patterns instead of general patterns.

The following cross-validation iterators were considered from which one was selected to be used.

- *k*-fold - This cross-validation iterator randomly divides the data into *k* groups of samples where all the samples. In total there are *k* splits where one of the groups is selected as the testing set for each split. In total the entire dataset will have been used in testing across all the splits. However, this cross-validation iterator does not solve either issue (i) or (ii) [21].

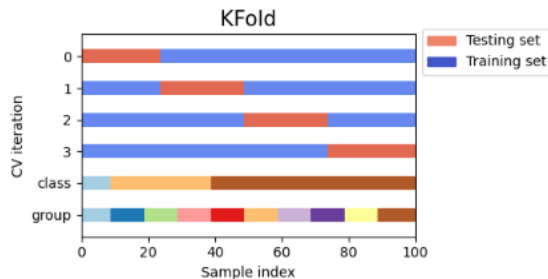


Fig. 1. How the data can be split in a GroupKFold. Source: [21]

- Group *k*-fold - This cross-validation iterator works in the same way as *k*-fold; however it ensures that the same group is not represented in both the training and testing set. The division of groups is made possible by providing an id with each data row. In the data used in this report, the id corresponds therefore to the `feats_group_uid`, meaning each patient. Therefore this cross-validation iterator takes into account issue (ii) but not (i). Due to an imbalance of the data the sizes of the training and testing sets in each split are not necessarily the same [21].

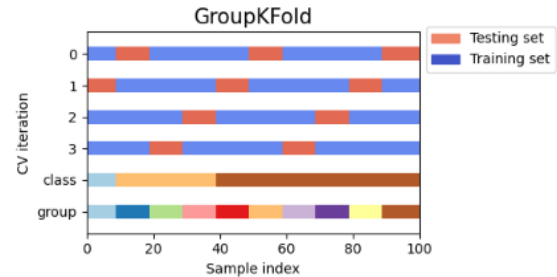


Fig. 2. How the data can be split in a GroupKFold. Source: [21]

- Stratified *k*-fold - This cross-validation iterator is a variation of the *k*-fold where each set in the splits contains approximately the same percentage of samples of each target class as the complete set. This means that the iterator tries to preserve the ratios of the classes in both the training and the test set. This cross-validation iterator solves the issue (i) but not (ii). Due to imbalances of the data, the ratios are not necessarily the same but given that it is possible to divide the data a split will not result in a set missing any positive labels [21].

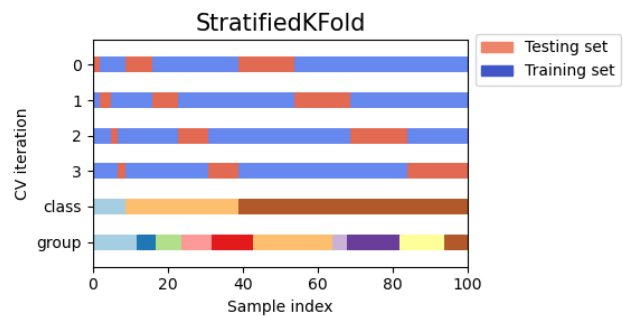


Fig. 3. How the data can be split in a StratifiedKFold. Source: [21]

- Stratified group *k*-fold - This cross-validation iterator combines both the group *k*-fold and stratified *k*-fold cross-validation iterators. Thus, this iterator preserves both the ratios of classes in each split and keeps each group within either the test or training split within a single split. This cross-validation iterator solves both issues (i) and (ii).

Finally, the best cross-validation iterator that caters to the data that does not provide methodological errors is the Stratified

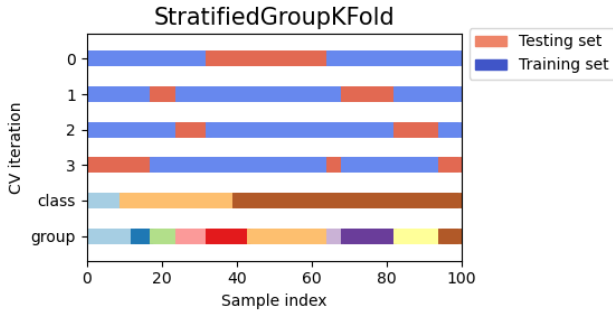


Fig. 4. How the data can be split in a StratifiedGroupKFold. Source: [21]

Group k -fold cross-validation iterator. Henceforth, when referring to a cross-validation iterator it is assumed that stratified group k -fold is used.

F. Model Evaluation for Highly Imbalanced Data

The four possible outcomes when the model makes a prediction are the following:

- 1) True Positive (TP): True positives occur when the model correctly predicts that a patient has sepsis.
- 2) True Negative (TN): True negatives occur when the model correctly predicts that a patient does not have sepsis.
- 3) False Positives (FP): False positives occur when the model incorrectly predicts that a patient has sepsis.
- 4) False Negatives (FN): False Negatives occur when the model incorrectly predicts that a patient does not have sepsis.

With these four outcomes, they can be used to determine the overall quality of the model. Typically, the following metrics are used [22].

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \quad (20)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (21)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (22)$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \quad (23)$$

$$\text{True Positive Rate} = \text{TPR} = \frac{\text{TP}}{P} \quad (24)$$

$$\text{False Positive Rate} = \text{FPR} = \frac{\text{FP}}{P} \quad (25)$$

However, due to the data being highly imbalanced, the accuracy would be high just by creating a model that predicts all inputs as non-sepsis-like. Given the distribution of classes in the provided data, such a model would have an accuracy of about 99.5% even if $\text{TP} = 0$. Moreover, these metrics are threshold-dependent, which is not of importance in the process of discriminating between sepsis-like and non-sepsis-like time windows. A better approach is thus to look at the receiver operating characteristic, which is received by plotting the true positive rate against the false positive rate. In other words,

the ROC curve shows the true positive rate for the classifier given an accepted value of the false positive rate. A random classifier follows a linear slope in which the true positive rate and the true negative rate are always equal. The area under the ROC, also called the ROC AUC, can be calculated to be used as a scoring metric. A ROC AUC of 0.5 indicates that there is no discrimination between classes, and 1 indicates perfect discrimination [22]. Henceforth, the ROC AUC will be used as the scoring metric in all machine learning models.

G. Hyperparameter Tuning

1) *Workflow*: An important part of training a machine learning model is to find the parameters that generate the best performance according to an evaluation method, which in this instance is the best ROC AUC. The workflow of finding the best hyperparameters consists of first defining a hyperparameter space meaning all potential values of hyperparameters that aims to be tested. Combinations from the hyperparameter space are tested together with a cross-validation iterator. A brute force solution would consist of testing all the possible combinations of parameters from the parameters space to select the best performing combination. After having found the retrained model, final cross-validation over the entire dataset is tested to find a final evaluation of the performance of the model.

The specific hyperparameters that are available to work within the parameter space are the machine learning model and the specific implementation in the library.

2) *Random Forest Hyperparameters*: The following hyperparameters for the random forest model are tuned according to their implementation in the sci-kit-learn library [18].

- *n estimators* - The number of trees in the forest. The default value is 100. Typical values to consider are integers ranging from 10 to 300 [18].
- *maximum depth* - The maximum depth of the tree. The default value is unlimited. Typical values to consider are integers ranging from 1 to 30 [18].
- *maximum features* - The number of features to consider when looking for the best. The default value is the same as the number of features which correspond to 24 for the provided dataset. Typical values to consider are integers ranging from 1 to the number of features = 24 [18].
- *minimum samples split* - The minimum number of samples required to split an internal node. The default value is 2. Typical values to consider are integers ranging from 2 to 30 [18].
- *minimum samples leaf* - The minimum number of samples required to be at a leaf node. The default value is 1. Typical values to consider are integers ranging from 1 to 30 [18].

3) *XGBoost Hyperparameters*: The following hyperparameters for the random forest model are tuned according to their implementation in the XGBoost library [19].

- *n estimators* - The number of trees [19]. Increasing this will make the model more complex and more likely to overfit. The default value is 100. Typical values to

consider in the hyperparameter space are integers ranging from 10 to 500 [19].

- **maximum depth** - Maximum depth of a tree. Increasing this value will make the model more complex and more likely to overfit. The default value is 6. Typical values to consider in the hyperparameter space are integers ranging from 1 to 10 [19].
- **learning rate** - Step size shrinkage used in the update to prevent overfitting. After each boosting step, weights of new features can be obtained directly, and the learning rate shrinks the feature weights to make the boosting process more conservative. The default value is 0.3. Typical values to consider in the hyperparameter space are values ranging from 0.01 to 1 in 0.01 step [19].
- **minimum child weight** - Minimum sum of instance weight (hessian) needed in a child. If the tree partition step results in a leaf node with the sum of instance weight less than minimum child weight, then the building process will give up further partitioning. The default value is 1. Typical values to consider in the hyperparameter space are values ranging from 0.5 to 1.5 in 0.01 steps [19].
- **gamma** - Minimum loss reduction required to make a further partition on a leaf node of the tree. The larger gamma is, the more conservative the algorithm is. The default value is 0. Typical values to consider in the hyperparameter space are values ranging from 0 to 0.5 in 0.01 steps [19].
- **column sample by tree** - The subsample ratio of columns when constructing each tree. Subsampling occurs once for every tree constructed. The default value is 1. Typical values to consider in the hyperparameter space are values ranging from 0.5 to 1 in 0.01 steps [19].
- **subsample** - subsample ratio of the training instances. Setting it to 0.5 means that XGBoost would randomly sample half of the training data before growing trees and this will prevent overfitting. Subsampling will occur once in every boosting iteration. The default value is 1. Typical values to consider in the hyperparameter space are values ranging from 0.5 to 1 in 0.01 steps [19].
- **lambda** - L2 regularization term on weights. Increasing this value will make the model more conservative, punishing complex models. The default value is 1. Typical values to consider in the hyperparameter space are values ranging from 0.5 to 1.5 in 0.01 steps [19].
- **alpha** - L1 regularization term on weights. Increasing this value will make the model more conservative, punishing complex models. The default value is 0. Typical values to consider in the hyperparameter space are values ranging from 0 to 0.5 in 0.01 steps [19].
- **scale positive weight** - Control the balance of positive and negative weights, useful for unbalanced classes. A typical value to consider: $\text{sum}(\text{negative instances}) / \text{sum}(\text{positive instances})$ [19].

4) *Hyperparameter spaces notation*: The following notation is used to define the entire hyperparameter space H of q

different parameters:

$$H = P_1 \times \cdots \times P_q = \bigotimes_i P_i \quad (26)$$

Each hyperparameter space P_i corresponds to hyperparameter i and consists of the set of possible hyperparameter values that can be chosen. The following notation for the sets is used:

$$[a, b, s] = \text{All real values from } a \text{ to } b \text{ (including } b) \text{ with step sizes of } s \text{ starting from } a \quad (27)$$

The size or cardinality of the hyperparameter space is given by $|H|$.

5) *The cardinality of initial hyperparameter spaces*: Using the notation in equations (26) and (27) the cardinality of the hyperparameter space that uses the entire range of typical values would be: $|H_{\text{Typical values Random Forest}}| \sim 10^8$ and $|H_{\text{Typical values XGBoost}}| \sim 10^{14}$. Assuming a generous computation time of 0.01 seconds for each cross-validation, an entire exhaustive search of the entire hyperparameter space would take about a month and 20 thousand years, respectively. An exhaustive search is therefore not an option.

6) *Hyperparameter Search*: To reduce the number of parameter combinations that are searched for from the initial total hyperparameter spaces $H_{\text{Typical values Random Forest}}$ and $H_{\text{Typical values XGBoost}}$ a combination of the following methods is used.

- **Cross validated grid search** - This method exhaustively goes through all the hyperparameter combinations in H and runs an iteration of cross-validation to calculate an average scoring value using an evaluator function which is ROC AUC [23].
- **Cross validated random halving grid search** - This method starts with selecting one "resource" parameter that has the property that it correlates with the increased resource usage of the model as it increases. For both the random forest and XGBoost a suitable hyperparameter is the n estimators. The minimum value and maximum values of the n estimators follow the limits of the set $P_{n \text{ estimators}}$. From $H \setminus P_{n \text{ estimators}}$ random values are sampled. During each iteration of the search, the cross-validated score of each chosen combination is calculated and the best third combinations are chosen to continue to the next iteration. This continues until there only exists one combination left. The number of candidates in the initial iteration is chosen so that the last iteration uses as many resources as possible within the limits of the maximum value for the "resource" parameter [24].

The chosen approach to finding the hyperparameter spaces in this project is a combination of, (1) a cross-validated random halving search to find a starting position using $H_{\text{Typical values Random Forest}}$ and $H_{\text{Typical values XGBoost}}$, (2) one or more narrow cross-validated grid searches with H s that use the previous best parameters $\pm \approx 10\%$ of range in $H_{\text{typical values}}$ for each P_i . The hyperparameter space is extended if the best parameter found is on the boundary of the H . In this step, only two to four hyperparameters are searched for in each iteration of the grid search using the previous best value when continuing with the next search.

This limitation is needed to reduce the cardinality of the H s. When finding the hyperparameters for the random forest all the grid searches are divided into two groups in the following order $[n \text{ estimators, maximum depth, maximum features}]$, $[\text{minimum samples split, minimum samples leaf}]$. The reasoning is that the first group mainly deals with the complexity of the model and the second group deals with node splits [18].

When finding the hyperparameters for the XGBoost model the grid searches are divided into three groups in the following order $[n \text{ estimators, maximum depth, learning rate, minimum child weight}]$, $[\text{gamma, column sample by tree, subsample}]$, $[\text{lambd, alpha}]$. The reasoning is that the first group mainly deals with general tree parameters, the second group deals with node construction, and the third group deals with general regularization parameters that penalize complex models. The regularization parameters are also left out during the initial cross-validated random halving search.

IV. RESULTS

A. Random Forest

1) *Before hyperparameter Tuning:* Using the default parameters of the random forest model before any tuning resulted in a mean ROC AUC of 0.67 with a standard error of 0.04 in the final cross-validation iteration. The ROC curve is presented in Figure5.

Receiver operating characteristic Random Forest before tuning hyperparameter

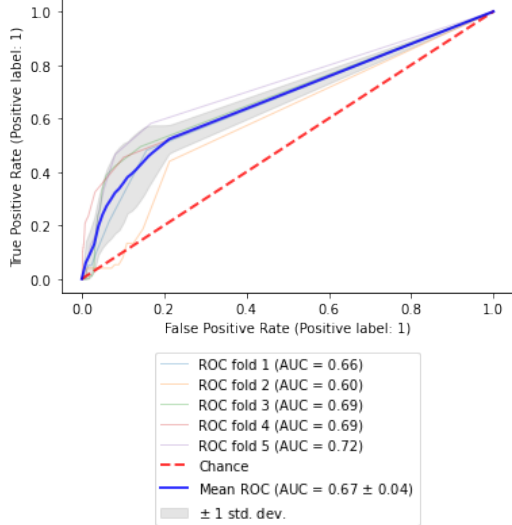


Fig. 5. Results of the Random Forests model before tuning for hyperparameters

2) *Final Hyperparameters:* The following hyperparameters were found as the best hyperparameters after following the tuning procedure described in the methods section. See the

appendix for a detailed description of all steps.

$$\begin{cases} n \text{ estimators} = 190 \\ \text{maximum depth} = 1 \\ \text{maximum features} = 2 \\ \text{minimum samples split} = 3 \\ \text{minimum samples leaf} = 23 \end{cases} \quad (28)$$

The final performance of the model received a mean ROC AUC of 0.842 with a standard error of 0.10 in the cross-validation iteration. The ROC curve is presented in Figure6.

Receiver operating characteristic Random Forest with Tuned Hyperparameters

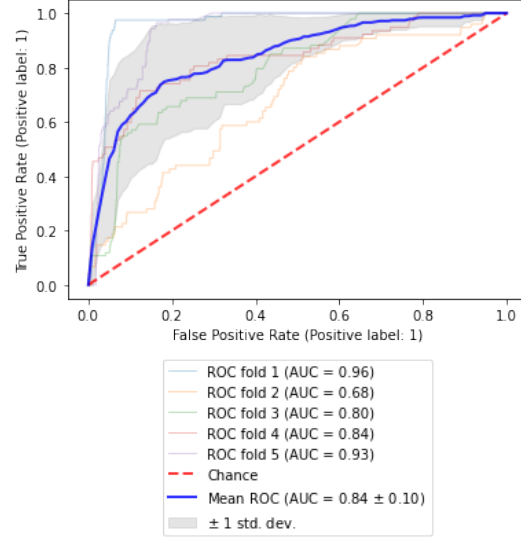


Fig. 6. Results of the Random Forests model after tuning for hyperparameters

B. Results XGBoost

1) *Before hyperparameter Tuning:* Using the default parameters of the XGBoost model before any tuning, except for the scale positive weight hyperparameter that, according to the typical convention, was set to $\text{sum(negative instances)} / \text{sum(positive instances)}$, which for the dataset was equal to 232.78, resulted in a mean ROC AUC of 0.75 with a standard error of 0.14 in the final cross-validation iteration. The ROC curve is presented in Figure5.

2) *Final hyperparameters:* The following hyperparameters were found as the best hyperparameters after following the tuning procedure described in the methods section. See the

Receiver operating characteristic XGBoost before tuning hyperparameters

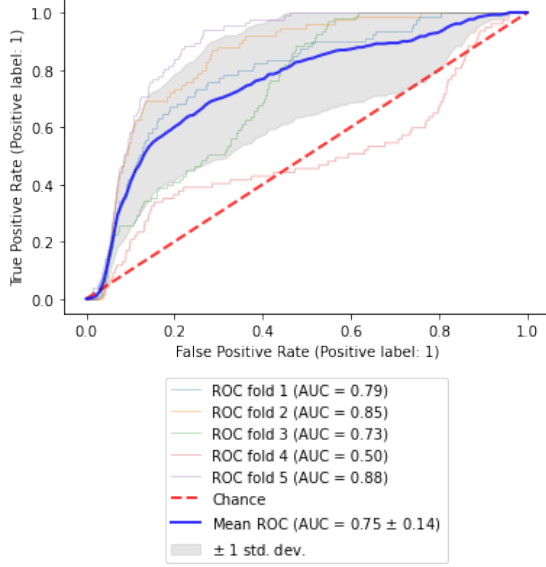


Fig. 7. Results of the XGBoost model before tuning for hyperparameters

Receiver operating characteristic XGBoost with tuned hyperparameters

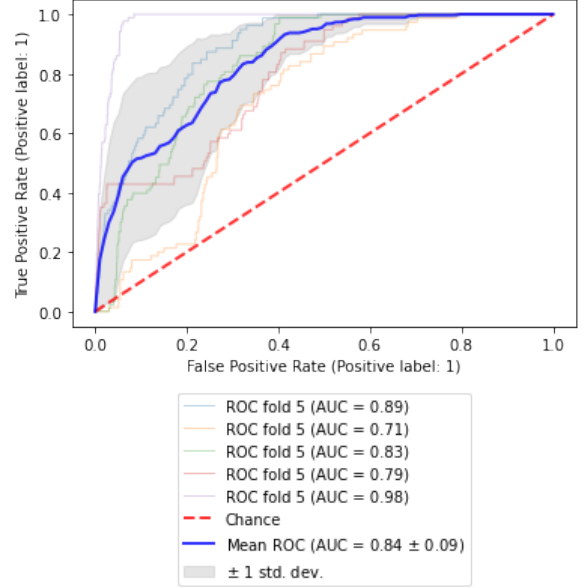


Fig. 8. Results of the XGBoost model after tuning for hyperparameters

appendix for a detailed description of all steps.

$$\left\{ \begin{array}{l} n \text{ estimators} = 140 \\ \text{maximum depth} = 1 \\ \text{learning rate} = 0.34 \\ \text{minimum child weight} = 0 \\ \text{gamma} = 0 \\ \text{column sample by tree} = 0.92 \\ \text{subsample} = 0.6 \\ \text{lambda} = 0.86 \\ \text{alpha} = 0.17 \\ \text{scale positive weight} = 232.78 \end{array} \right. \quad (29)$$

The final performance of the model received a mean ROC AUC of 0.840 with a standard error of 0.10 in the cross-validation iteration. The ROC curve is presented in Figure 8.

V. DISCUSSION

A. Model Performance

The final random forest model demonstrated a mean ROC AUC $\overline{AUC} = 0.84$ with a standard deviation $s = 0.1$ and the final XGBoost model demonstrated a mean ROC AUC $\overline{AUC} = 0.84$ with a standard deviation $s = 0.09$. Any value for the ROC AUC that is greater than 0.5 indicates that the model has the ability to distinguish between sepsis-like and non-sepsis-like time windows [22]. It is natural to perform a hypothesis test to evaluate whether \overline{AUC} differs significantly from 0.5. The test statistic given by $(\overline{AUC} - 0.5)/d$ is approximately normally distributed [25]. The standard error $d = s/\sqrt{n}$ where $n = 5$ is equal to the number of folds in the cross-validation iterator. With the null and alternative hypotheses defined as $H_0 : \overline{AUC} = 0.5$ versus $H_1 : \overline{AUC} \neq 0.5$ yields a test statistics with p -values of $7 \cdot 10^{-12}$ and $9 \cdot 10^{-14}$ for the random forest model and XGBoost model respectively. Thus,

significantly rejecting the null hypothesis, showing that both models can distinguish sepsis-like windows. The grade of a

TABLE II
GRADES OF DIAGNOSTIC TESTS ACCORDING TO THE ROC AUC [25]

ROC AUC	Grade
0.9 - 1	Outstanding
0.8 - 0.9	Excellent
0.7 - 0.8	Acceptable
0.6 - 0.7	Poor
≤ 0.6	Unacceptable

diagnostic test predictor in medicine can be defined according to the categories in table II [25]. Testing whether a model at least has a specific grade g can be done by formulating the null hypotheses $H_0 : \overline{AUC} < l_g$ against the alternative hypothesis $H_1 : \overline{AUC} \geq l_g$ where l_g is the lower limit in the ROC AUC for the grade g . The p values for models that at least achieve different grades are given in Table III. It is shown

TABLE III
P VALUES FOR THE MODELS TO AT LEAST HAVE A SPECIFIC GRADE

At least ...	XGBoost p -value	Random forest p -value
Outstanding	0.91	0.88
Excellent	0.19	0.20
Acceptable	$1.1 \cdot 10^{-2}$	$2.2 \cdot 10^{-2}$
Poor	$7.1 \cdot 10^{-8}$	$6.0 \cdot 10^{-7}$

that both the random forest and the XGBoost models with significance have at least an acceptable grade as a diagnostic test. Although measures have been taken to reduce bias and overfitting, it is still not guaranteed that the model performs at the calculated level. Only patients from Sweden have been used in the data set, which could have introduced selection bias. Introducing a validation set would have provided more information on overfitting. A common denominator regarding

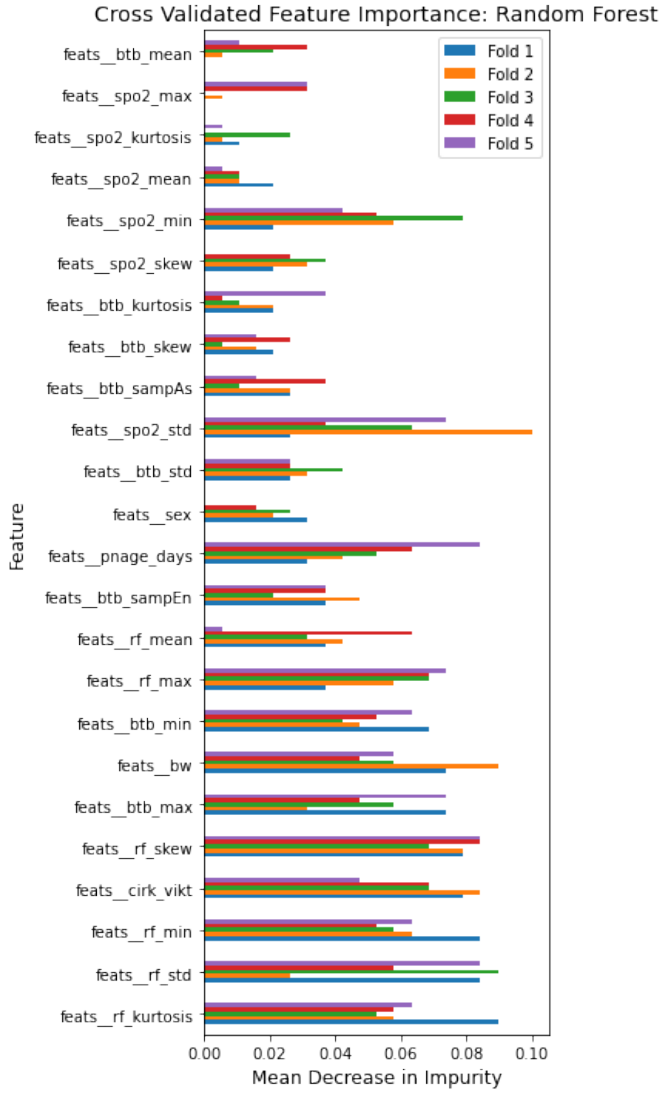


Fig. 9. Feature importance of the Random Forest model

the pitfalls of the results is that they could be partially eluded if the data set had been larger.

1) *Comparison with the SOFA score:* The SOFA score has a highly distinctive ability to predict sepsis with a ROC AUC of 0.89 [26]. However, machine learning models that are close to that level would still be of interest, especially if they do not require invasive methods and increased tasks for healthcare personnel, as in the case of the SOFA score [4].

B. Feature Importance

Feature importances for the final random forest and XGBoost models were calculated by finding the mean decrease in impurity when a feature is removed. The (Gini) impurity is a metric similar to entropy from information theory. A large decrease when a feature is removed signifies greater importance. Calculations were performed on a cross-validation iteration to get a sense of whether the importances were random or not. The importance of each parameter is found in Figures 9 and 10.

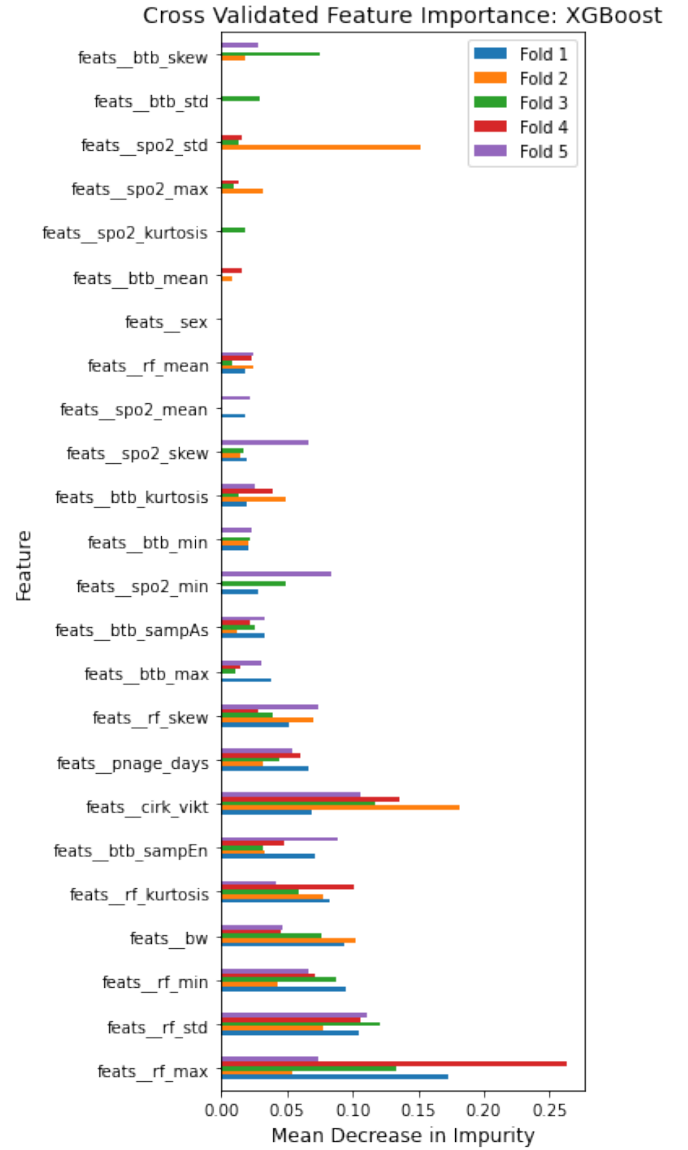


Fig. 10. Feature importance of the XGBoost model

1) *Insights:* For both models, the feature importances for the different folds fluctuated, indicating a component of randomness within the models. However, it is possible to find some general patterns that apply to both models.

- Features of high importance - All features regarding the respiratory frequency and body weight.
- Features of low importance - All features regarding oxygen saturation and sex.
- Features with varying importance - The beat-to-beat interval features achieved varying importance with the maximum and sample entropy features scoring on the higher side within the group.

The results go somewhat against the SOFA score that takes into account oxygen saturation, which was shown to be less important, but none of the respiratory frequencies, bodyweight measurements, or beat-to-beat intervals were shown to be important [11].

C. Model Complexity

Both the random forest and XGBoost models ended up with hyperparameters toward the less complex side, the most significant hyperparameter being a maximum depth of only 1. This signifies that only one comparison of one feature is carried out in every tree. It is up for discussion whether a single node even should be called a tree. An analysis of the model performance dependence on hyperparameter values was performed using data from the hyperparameter tuning procedure. Some of the discovered relationships are as follows.

- Keeping all other hyperparameters the same, the performance of the XGBoost model was greatly reduced with depths of 5 having ROC AUCs of 0.65 to 0.7, while depths of 1 and 2 performed between 0.8 and 0.85.
- Keeping all other hyperparameters the same, the performance of the random forest model was greatly reduced for maximum features greater than 20 with a ROC AUC of 0.65 to 0.7, while depths of maximum features less than 5 performed closer to 0.8.
- For both models, performance plateaued for n estimators larger than 150.

See the appendix for more relationships and graphs of the dependence of the ROC AUC score on the hyperparameters for both the training and the testing sets.

1) *Implications for explainability:* Since both models tended towards the less complex side, a relevant question is whether the models chosen are too complex. On the one hand, it can be argued that there exists room for adding more features and finding more complex patterns. On the other hand, one could see the possible implications for explainability. It is highly debated whether artificial intelligence in healthcare should be explainable, which in essence means that a human should understand why the model makes its decisions [27]. In the current state of the world, it is easier to overcome legal and ethical hurdles with explainable models [27]. Together with the fact that simpler models tend to be more explainable, this implies that having a simple model that is the best performing model brings hope for easy adoption [28]. Since both the random forest and XGBoost are built on the concept of a decision tree that asks simple questions at each node, it is theoretically possible to reverse engineer the decision tree and come up with an easier set of questions that could replace the SOFA scoring system.

VI. CONCLUSION

In conclusion, the use of random forests and XGBoost for the detection of neonatal sepsis is feasible, offering performance within the range of the SOFA scoring system. At the same time, requiring less invasive methods and being automated in the sense that healthcare personnel is not burdened with more tasks. However, both models tended to perform better with reduced complexity. This suggests that it could be possible to gain benefit from adding more features, thus discovering more complex patterns in the data. Or else generate even simple models that focus on the explainability of the data to ease the adoption of the algorithm. One possible method would be to reverse engineer the questions asked at

specific nodes in the decision tree ensemble models.

The project has also found appropriate methods for validation and evaluation that could be generalized to all types of highly unbalanced grouped tabular data that can generalize to other similar applications.

A. Future Work

A possible extension of this project is to calibrate the final models. Currently, due to the usage of non-threshold-dependent scoring systems, the model does not find the exact probability of input being sepsis-like. Generating the probabilities from the model predictions would require calibration with prior probabilities [7]. Another extension is to reverse engineer the models to find simple questions that enhance explainability. Other future work includes acquiring a larger dataset with more sample points and features and testing other diseases or conditions.

ETHICS STATEMENT

All patient data were anonymized by removing personal identification numbers and not providing the exact date and time of the collected data. Therefore, precluding any attempt to identify specific patients. Accessing the data required multi-factor authentication and the generation of one-time passwords for each login session.

ACKNOWLEDGMENT

The authors would like to express gratitude to Antoine Honoré for his support and assistance throughout the course of the project. Furthermore, expressing a special appreciation for the Herlenius Research Team for their work in collecting, working with, and providing the dataset for this project.

SOURCE CODE

The source code for the project and all plots can be found in the following GitHub repository: <https://github.com/marwanbardaji/Neonatal-Sepsis-Detection-Using-Decision-Tree-Ensemble-Methods-Random-Forest-and-XGBoost>.

REFERENCES

- [1] A.-M. Audet, S. Greenfield, and M. Field, "Medical practice guidelines: current activities and future directions," *Annals of Internal Medicine*, vol. 113, no. 9, pp. 709–714, 1990.
- [2] M. Singer, C. S. Deutschman, C. W. Seymour, M. Shankar-Hari, D. Annane, M. Bauer, R. Bellomo, G. R. Bernard, J.-D. Chiche, C. M. Coopersmith, R. S. Hotchkiss, M. M. Levy, J. C. Marshall, G. S. Martin, S. M. Opal, G. D. Rubenfeld, T. van der Poll, J.-L. Vincent, and D. C. Angus, "The Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3)," *JAMA*, vol. 315, no. 8, pp. 801–810, 02 2016. [Online]. Available: <https://doi.org/10.1001/jama.2016.0287>
- [3] C. Fleischmann, F. Reichert, A. Cassini, R. Horner, T. Harder, R. Markwart, M. Tröndle, Y. Savova, N. Kissoon, P. Schlattmann *et al.*, "Global incidence and mortality of neonatal sepsis: a systematic review and meta-analysis," *Archives of Disease in Childhood*, vol. 106, no. 8, pp. 745–752, 2021.
- [4] E. P. Raith, A. A. Udy, M. Bailey, S. McGloughlin, C. MacIsaac, R. Bellomo, D. V. Pilcher, for the Australian, N. Z. I. C. S. A. C. for Outcomes, and R. E. (CORE), "Prognostic Accuracy of the SOFA Score, SIRS Criteria, and qSOFA Score for In-Hospital Mortality Among Adults With Suspected Infection Admitted to the Intensive Care Unit," *JAMA*, vol. 317, no. 3, pp. 290–300, 01 2017. [Online]. Available: <https://doi.org/10.1001/jama.2016.20328>

- [5] B. Cummings, "Rising healthcare costs are a rising concern," *Journal of Financial Planning*, vol. 35, no. 2, pp. 19–19, 2022.
- [6] M. Marć, A. Bartosiewicz, J. Burzyńska, Z. Chmiel, and P. Januszewicz, "A nursing shortage—a prospect of global and local policies," *International nursing review*, vol. 66, no. 1, pp. 9–16, 2019.
- [7] R. N. Stuart Peter, *Artificial Intelligence A Modern Approach*, 3rd ed. Harlow, England: Pearson Education Limited, 2016.
- [8] A. Honore, D. Liu, D. Forsberg, K. Coste, E. Herlenius, S. Chatterjee, and M. Skoglund, "Hidden markov models for sepsis detection in preterm infants," in *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020* :, ser. International Conference on Acoustics Speech and Signal Processing ICASSP. Institute of Electrical and Electronics Engineers (IEEE), 2020, pp. 1130–1134, qC 20210324.
- [9] A. Honoré, D. Forsberg, K. Jost, K. Adolphson, A. Stålhammar, E. Herlenius, and S. Chatterjee, "Classification and feature extraction for neonatal sepsis detection," 2022.
- [10] K. D. Fairchild and T. M. O'Shea, "Heart rate characteristics: physiometers for detection of late-onset neonatal sepsis," *Clinics in perinatology*, vol. 37, no. 3, pp. 581–598, 2010.
- [11] A. Oscarson, C. Bjurman, J. E. Wallér, and M. Werner, "Sepsis hos vuxna – tidig upptäckt och initial behandling," *Läkartidningen*, Mar 2017.
- [12] A. Rhodes, L. E. Evans, W. Alhazzani, M. M. Levy, M. Antonelli, R. Ferrer, A. Kumar, J. E. Sevransky, C. L. Sprung, M. E. Nunnally *et al.*, "Surviving sepsis campaign: international guidelines for management of sepsis and septic shock: 2016," *Intensive care medicine*, vol. 43, no. 3, pp. 304–377, 2017.
- [13] P.-Y. Iroh Tam and C. M. Bendel, "Diagnostics for neonatal sepsis: current approaches and future directions," *Pediatric research*, vol. 82, no. 4, pp. 574–583, 2017.
- [14] J. L. Wynn and R. A. Polin, "A neonatal sequential organ failure assessment score predicts mortality to late-onset sepsis in preterm very low birth weight infants," *Pediatric research*, vol. 88, no. 1, pp. 85–90, 2020.
- [15] M. J. Aspinall, "Use of a decision tree to improve accuracy of diagnosis," *Nursing Research*, vol. 28, no. 3, pp. 182–185, 1979.
- [16] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [17] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 785–794. [Online]. Available: <https://doi.org/10.1145/2939672.2939785>
- [18] SKlearn, "Sklearn.ensemble.randomforestclassifier," 2022. [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
- [19] XGBoost, "Xgboost parameters," 2021. [Online]. Available: <https://xgboost.readthedocs.io/en/stable/parameter.html>
- [20] J. Brownlee, "Train-test split for evaluating machine learning algorithms," Aug 2020. [Online]. Available: <https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/>
- [21] Scikit, "3.1. cross-validation: Evaluating estimator performance," 2022. [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html
- [22] X.-H. Zhou, D. K. McClish, and N. A. Obuchowski, *Statistical methods in diagnostic medicine*. John Wiley & Sons, 2009.
- [23] SKlearn, "Sklearn.model_selection.gridsearchcv," 2022. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
- [24] Scikit, "Sklearn.model_selection.halvingrandomsearchcv," 2022. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.HalvingRandomSearchCV.html
- [25] J. N. Mandrekar, "Receiver operating characteristic curve in diagnostic test assessment," *Journal of Thoracic Oncology*, vol. 5, no. 9, pp. 1315–1316, 2010.
- [26] A. K. Toker, S. Kose, and M. Turken, "Comparison of sofa score, sirs, qsofa, and qsofa+ 1 criteria in the diagnosis and prognosis of sepsis," *The Eurasian Journal of Medicine*, vol. 53, no. 1, p. 40, 2021.
- [27] J. Amann, A. Blasimme, E. Vayena, D. Frey, and V. I. Madai, "Explainability for artificial intelligence in healthcare: a multidisciplinary perspective," *BMC Medical Informatics and Decision Making*, vol. 20, no. 1, pp. 1–9, 2020.
- [28] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, "Explainable ai: A review of machine learning interpretability methods," *Entropy*, vol. 23, no. 1, p. 18, 2020.

Neonatal Sepsis Detection With Random Forest Classification for Heavily Imbalanced Data

Ayman Osman Abubaker

Abstract—Neonatal sepsis is associated with most cases of mortality in the neonatal intensive care unit. Major challenges in detecting sepsis using suitable biomarkers has lead people to look for alternative approaches in the form of Machine Learning techniques. In this project, Random Forest classification was performed on a sepsis data set provided by Karolinska Hospital. We particularly focused on tackling class imbalance in the data using sampling and cost-sensitive techniques. We compare the classification performances of Random Forests in six different setups; four using oversampling and undersampling techniques; one using cost-sensitive learning and one basic Random Forest. The performance with the oversampling techniques were better and could identify more sepsis patients than the other setups. The overall performances were also good, making the methods potentially useful in practice.

Sammanfattning—Neonatal sepsis är orsaken till majoriteten av mortaliteten i neonatal intensivvården. Svårigheten i att detektera sepsis med hjälp av biomarkörer har lett många att leta efter alternativa metoder. Maskininlärningstekniker är en sådan alternativ metod som har i senaste tider ökat i användning inom vård och andra sektorer. I detta projekt användes Random Forest klassifikations algoritmen på en sepsis datamängd given av Karolinska Sjukhuset. Vi fokuserade på att hantera klass imbalance i datan genom att använda olika provtagnings- och kostnadskänsliga metoder. Vi jämförde klassificeringsprestanda för Random Forest med sex olika inställningar; fyra av de använde provtagningsmetoderna; en av de använde en kostnadskänslig metod och en var en vanlig Random Forest. Det visade sig att modellens prestanda ökade som mest med översamlings metoderna. Den generella klassificeringsprestandan var också bra, vilket gör Random Forests tillsammans med provtagningsmetoderna potentiellt användbar i praktiken.

Index Terms—Random Forest, Neonatal Sepsis, Imbalanced Classification, Cost-sensitive, SMOTE, ADASYN, CNN, Tomek-Links.

Supervisors: Antoine Honoré

TRITA number: TRITA-EECS-EX-2022:176

I. INTRODUCTION

Neonatal sepsis is a life threatening condition associated with most cases of mortality and morbidity in the neonatal intensive care unit (NICU) [1]. Early antibiotic treatment is known to improve the outcome of sepsis. However, early diagnosis of neonatal sepsis is difficult in practice [2]. Thus, early detection of neonatal sepsis is an important task in neonatal medicine.

A major challenge to early detection of sepsis stems from its non-specific signs and symptoms which are shared by disease states such as inflammation [3]. Considerable efforts have been made to identify suitable biomarkers that can aid the detection

of sepsis. But this has proven difficult mainly because of the lack of specificity and sensitivity [3]. Although there is no definitive diagnostic test for neonatal sepsis, laboratory testing relying on invasive tests such as blood tests can provide strong clues to whether a patient is developing sepsis. However, this is not time efficient and can delay early treatment. [4]

Another approach that has gained more popularity in recent years is the use of machine learning (ML) algorithms to exploit the ever-increasing amount of patient clinical data available in hospital wide databases [5]. These algorithms learn from collected data of previous patients to predict whether a new patient will develop sepsis based on a set of measurements such as monitoring signals and demographic information [3]. This can be done in a binary classification framework. There are several ML algorithms to handle binary classification problems. For sepsis detection problems in NICU, the algorithms rely on clinical data extracted from fixed duration time segments to produce a predicted probability of a patient developing sepsis [6].

Linear binary classifiers are usually chosen to obtain a risk score that clinicians can interpret. But it is still unclear whether non-linear and interpretable algorithms, such as Random Forests, can be useful to improve classification results.

In this project, we aim to examine the classification results of the Random Forest model applied to a sepsis data set provided by Karolinska Hospital. The following research questions will be answered:

- 1) How does the classification performance vary when dataset sampling techniques are used over cost-sensitive techniques to handle the imbalanced data set?

II. THEORY

A. Time series data for classification problems

Time series data is a sequence of data points for a single subject collected at different points in time. At each time stamp, the measurements for a set of independent variables called features are collected from the subject. The corresponding dependant variable, referred to as the target, is also measured and assigned a value. In classification problems, this value is categorical and divided into class labels.

B. Random Forest

To understand the Random Forest algorithm, we have to first understand what decision trees are. A decision tree is a classifier which recursively splits a data set into smaller subsets. Each node is split into 2 child nodes, and the splitting

criterion is dictated by a "test" on the features [7]. For example, if age is one of the features, then a test could be to split the data set based on which samples are older than 20 years old. The nodes are recursively split in this way until a leaf node is reached. The leaf node is represented by the class label the majority of the samples in the belong to [7]. When a decision tree model is trained, the splitting criterion's are chosen suitably and the pathway from root to leaf is created. This pathway represents the classification rules of the tree [7].

Random Forests are an ensemble of decision trees. Each tree uses a random subset of the features for the splitting criterion's, and the final decision is based on a majority vote of all the trees. The benefits of Random Forests over decision trees is that it prevents overfitting [8].

C. Imbalanced data

A classification data set with skewed class proportions is called imbalanced. Most classifiers trained on imbalanced data are biased in their prediction, i.e. they are better at predicting the majority classes than they are at predicting the minority classes. [9] This could be a problem when predicting the minority classes are important for the application at hand. There are several methods to deal with imbalanced classification problems. The main methods used in this project and in general are sampling techniques [9] and cost-sensitive techniques [10] combined with appropriate evaluation metrics.

D. Cost-sensitive learning

Cost-sensitive learning is a type of learning that amplifies the cost of misclassifying a minority class [10]. In Random Forest, this can be done by assigning weights to each class which results in a biased splitting that increases the proportion of leaf nodes represented by the minority class [10].

E. Sampling techniques

Synthetic Minority Oversampling Technique (SMOTE) is an oversampling technique that generates new synthetic samples of a minority class by utilizing existing samples [11]. The algorithm iterates over every minority class sample. Once a sample is selected, SMOTE identifies its k nearest neighbors based on their Euclidean distance and at random selects a given number of those neighbors to use for the generation process. Each neighbor, together with the selected sample, generates a new synthetic sample in the following manner [11]:

- 1) Take the difference between the feature vector of the neighbor and the selected sample.
- 2) Multiply the answer with a random number between 0 and 1.
- 3) Add the answer to the feature vector of the selected sample.

In other words, SMOTE creates synthetic samples by choosing a random point along the line segment between two neighboring feature vectors of the same class. This effectively makes the decision boundary more general [11].

The Adaptive Synthetic Sampling (ADASYN) technique is another oversampling method. It is very similar to SMOTE,

but the key difference is that SMOTE lets every minority class sample generate an equal amount of synthetic samples whereas ADASYN prioritizes allowing harder to learn minority class samples to generate more samples [12]. ADASYN achieves this by using a density distribution function \hat{r}_i . For every minority class sample x_i , ADASYN finds the k -nearest neighbors and computes the following ratio:

$$r_i = \delta_i / k, \quad (1)$$

where δ_i is the number of majority class samples among the k nearest neighbors. Afterwards, the ratio is normalized to become a density distribution:

$$\hat{r}_i = \frac{r_i}{\sum_i r_i}, \quad (2)$$

If we want to upsample the minority class by a total value G , then the amount of synthetic samples, s_i , minority class sample, x_i , must generate is:

$$s_i = \hat{r}_i \times G \quad (3)$$

In other words, the minority class samples with many neighboring majority class samples will generate more samples. These minority class samples are harder to learn because they are either noise or closer to the decision boundary [12].

Condensed Nearest Neighbor (CNN) is an undersampling technique which reduces the size of the majority class. The underlying principle behind it is simple. CNN constructs a subset of majority samples which are able to correctly classify the original data set using a 1-NN algorithm [13]. Let U be the set of all minority class samples and X the set of all majority class samples. CNN algorithm then works iteratively as follows:

- 1) Remove a random sample from X and add it to U
- 2) Iterate over all elements x_i in X .
- 3) If the nearest neighbor of x_i in set U has a different class label than x_i , remove x_i from X and add it to U .
- 4) Repeat steps 2-4 until no more samples can be added to U .

This algorithm was created to reduce the memory requirements of kNN [13] but it also works effectively as an undersampling technique for imbalanced classification.

Tomek Links is another undersampling technique. It under-samples under a very simple rule. Whenever a majority class sample and a minority class sample are each other's nearest neighbor, Tomek Links will remove the majority class sample [14]. By reducing unwanted overlap between classes, Tomek Links reduces both noise and ambiguous points along the class boundary [14].

F. Evaluation metrics

When training and evaluating a ML model for imbalanced classification, it is important to choose appropriate metrics. For instance the total correctly classified samples, or in other words the accuracy, tells us nothing about how many correctly classified minority class samples there are. Examples of more insightful measurements are the True Positives (TP), True

Negatives (TN), False Positives (FP) and False Negatives (FN) [15]. Since this project focuses on binary classification problems, we denote the positive as the minority class and the negative as the majority class. The 4 measurements are thus defined as follows:

- TP = Number of positive samples correctly classified
- TN = Number of negatives samples correctly classified
- FP = Number of negative samples misclassified
- FN = Number of positive samples misclassified

Using these 4 measurements, we can define other measurements independant of the sample size:

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

$$Precision = \frac{TP}{TP + FP}, \quad (5)$$

$$F1 = 2 \times \frac{Recall \times Precision}{Recall + Precision}, \quad (6)$$

Intuitively, recall is the ability of a model to correctly classify positive samples, precision the ability to not misclassify a negative sample, and F1 the harmonic mean between them [15]. These 3 performance metrics are good for distinguishing between classifiers when dealing with imbalanced data [15]. They all reach an optimum value at 1.

Another useful performance metric is the area (AUC) under the receiver operating characteristic curve (ROC). The ROC is a plot of the recall against the False Positive Rate (FPR is defined below) for all possible cut-off points [15]. Cut-off represents the minimum threshold after which a predicted probability would be classified as a positive sample.

$$FPR = \frac{FP}{FP + TN} \quad (7)$$

AUC is intuitively the ability of the model to distinguish between the classes. If AUC is 0.5, then the model can make no distinction between the two classes, whereas a value of 1 is a perfect distinction.

G. Cross-validation

A model that performs well on training data but generalizes poorly is called an over fit model. There are many measures taken to prevent over fitting and one of them is using a cross-validation scheme. Cross-validation is a re-sampling method that uses different parts of the data to evaluate and train a model on different iterations. By evaluating and training on different parts of the data, we ensure that the model does not over fit to a specific part of the data and it gives us a way to estimate the error of the performance metrics [16].

K -fold cross-validation is a commonly used method. It divides the data into k number of folds, and in each iteration 1 fold is used for evaluation and the remaining $k - 1$ folds for training. The performance metrics are computed for all the iterations and the average is taken as the final model performance [16].

H. Calibration

When we are interested in the predicted probability of a positive sample such as in this project, it is important to calibrate the probabilities. Calibration ensures that the predicted probability actually matches the true likelihood of the event. This also aids in defining cut-offs accordingly [17].

When calibrating a model, we fit a new model on the old model output to scale its predicted probability and thus require a separate calibration set. This data set will not be used for the training in order to prevent bias in the calibration [18].

III. METHOD

A. Data set and data pre-processing

The data set used in this project was provided by Karolinska Hospital. It is a combination of patient demographics information and time series data obtained by monitoring the vital parameters of 108 patients, 10 of which develop sepsis. The time series data was divided into intervals of around one hour. For each interval, 24 features and 2 class labels (1 if patient was diagnosed as having sepsis, 0 otherwise) were extracted. If the patient was diagnosed as having sepsis, all the samples obtained within 24 hours before the time of diagnosis were labeled as septic. In total, there are 134668 data points, of which only 556 samples are labeled as septic, which means the data set is highly imbalanced.

Most ML models including Random Forest are based on the assumption that the data are independant and identically distributed random variables (i.i.d). Therefore, despite the data set being a time series data, we assumed i.i.d for practical purposes.

The first step in the pre-processing was to remove the time stamps from the data set. Then the data was split in a 80:20 train to test ratio, making sure that the split was based on the patient id, and keeping the proportion of class labels constant. This resulted in a test set which had 20% of the total sepsis patients, 21% of the total sample points with class label 1 and 20% of the total data points.

The next step was to identify any missing values and replace them using an imputation method. We chose 5-NN because it was widely considered a good method to impute missing values. 5-NN estimates the missing values in a sample with the average of the values of the sample's 5 nearest neighbors. Imputation was done separately for the train and test sets because we did not want to leak any data from the train set to the test set because doing so would affect the generalization performance of the final model.

B. Training

In total, we trained using 6 different setups for the Random Forest model, 4 using the sampling techniques described in the theory section, 1 using cost-sensitive learning and 1 basic Random Forest. During the training phase, we performed a 5-fold cross validation on the training set. The splits were again based on patient id, with each fold having at least 1 patient with sepsis. Among the 5 folds, 1 was used for validation and the remaining 4 were used for training and calibration.

TABLE I
CROSS-VALIDATION SCORES FOR THE 6 DIFFERENT SETUPS

Cross-validation scores				
Setup	Precision	Recall	F1	AUC
Basic	0.296	0.109	0.069	0.853
Weighted	0.015	0.173	0.028	0.892
SMOTE	0.054	0.377	0.087	0.829
ADASYN	0.057	0.384	0.090	0.833
CNN	0.016	0.223	0.027	0.786
TomekLinks	0.256	0.107	0.066	0.856

TABLE II
FINAL MODEL PERFORMANCE ON TEST SET

Generalization performance				
Setup	Precision	Recall	F1	AUC
ADASYN	0.018	0.043	0.025	0.836

For each of the 6 setups, 100 Random Forest models with random combinations of hyper parameters were fitted, calibrated and evaluated. The chosen evaluation metric was recall because we were mainly interested in classifying the sepsis patients correctly.

After we obtained the optimized models for the respective setups, we measured the mean cross-validation scores for recall, AUC, precision and F1-score. The final model was chosen based on having the highest recall value and its generalization performance was evaluated using the test set.

IV. RESULTS

For SMOTE and ADASYN, the minority class samples were over sampled until a ratio 1:2 was obtained. For the cost sensitive learning setup, the class weights were inversely proportional to the corresponding class frequencies in the training sample.

The cross-validation scores for the 6 different setups are recorded in Table I. As we can see, Random Forest with ADASYN sampling technique achieved the highest recall value of 0.384 and was thus chosen as the final model. Its performance on the test set can be seen in Table II.

V. DISCUSSION

The cross-validation scores for all the setups have a good AUC. In general, an AUC between 0.8-0.9 is considered to have an excellent discrimination. However, as we see, the recall for the setups is not high enough to be considered excellent. A recall of 0.4 means that 60% of the positive samples were misclassified. In practice, this means that the algorithm fails to diagnose a sepsis patient 60% of the time.

Lowering the cut-off point could have improved the recall, but this would come at the expense of decreasing the precision. This is also something we observe in Table I. For SMOTE and ADASYN, the recall improved greatly compared to a basic Random Forest, but the precision likewise worsened. This trade-off between recall and precision occurs because to increase the recall, the model has to be more sensitive to

identify a positive sample. But this also increases the likelihood of misclassifying a negative samples which decreases the precision. It eventually becomes a question of how much precision we are willing to sacrifice up for a better recall.

Overall, the oversampling methods had a stronger effect on the performance than the undersampling methods. One reason for this could be that the undersampling methods CNN and Tomek Links have a limitation on how much they can reduce the majority class, so there will still be a class imbalance after implementing them. A way to avoid this could be to combine the methods instead of using them separately.

As we can see in Table II. the generalization performance of the best performing model with ADASYN was pretty good and reflected the training scores. This most likely means that the model did not over fit to the training data.

VI. CONCLUSION

In conclusion, the oversampling methods improved the classification results better than the undersampling methods and cost-sensitive technique. The generalization performance was also good, which shows that Random Forests in combination with oversampling techniques can potentially have useful applications in practice.

ACKNOWLEDGMENT

I would like to thank my supervisor Antoine Honoré for his support and guidance during the course of the project.

REFERENCES

- [1] C. Hornik, C. P. Fort, P., "Early and late onset sepsis in very-low-birth-weight infants from a large group of Neonatal Intensive Care Units," *Early Human Development*, vol. 88, pp. S69–S74, May 2012.
- [2] F. Masino, A. J., Harris, M.C., "Machine learning models for early sepsis recognition in the neonatal intensive care unit using readily available electronic health record data," *PLOS ONE*, vol. 14, Feb. 2019.
- [3] J. Moor, M., Rieck, B., "Early prediction of sepsis in the ICU using Machine Learning: A Systematic Review," *Frontiers in Medicine*, vol. 8, Feb. 2021.
- [4] N. Laforgia, B. Coppola, and R. Carbone, "Rapid detection of neonatal sepsis using polymerase chain reaction," *Acta Paediatrica*, vol. 86, p. 1097–1099, 1997.
- [5] P. Giacobbe, D. R., Signori, A., "Early Detection of Sepsis With Machine Learning Techniques: A Brief Clinical Perspective. Frontiers in medicine," *Frontiers in Medicine*, vol. 8, Feb. 2021.
- [6] C. Song, W., Jung, S. Y., "A Predictive Model Based on Machine Learning for the Early Detection of Late-Onset Neonatal Sepsis: Development and Observational Study," *JMIR medical informatics*, vol. 8, Jul. 2020.
- [7] X. Li, M. Li, Y. Zhang, and X. Deng, "A new random forest method based on belief decision trees and its application in intention estimation," in *2021 33rd Chinese Control and Decision Conference (CCDC)*, May 2021, pp. 6008–6012.
- [8] A. Kouzani, S. Nahavandi, and K. Khoshmanesh, "Face classification by a random forest," in *TENCON 2007 - 2007 IEEE Region 10 Conference*, Nov. 2007, pp. 1–4.
- [9] Z. Yuan and P. Zhao, "An improved ensemble learning for imbalanced data classification," in *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, May 2019, pp. 408–411.
- [10] X. Liu, "A benefit-cost based method for cost-sensitive decision trees," in *2009 WRI Global Congress on Intelligent Systems*, vol. 3, Aug. 2009, pp. 463–467.
- [11] H. Chawla, N. V., Bowyer, K. W., "Smote: Synthetic Minority over-Sampling Technique," *Journal of Artificial Intelligence Research*, p. 321–357, Jun. 2002.

- [12] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, Jun. 2008, pp. 1322–1328.
- [13] P. Hart, "The condensed nearest neighbor rule (corresp.)," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 515–516, May 1968.
- [14] G. Yang and L. Qicheng, "An over sampling method of unbalanced data based on ant colony clustering," *IEEE Access*, vol. 9, pp. 130 990–130 996, Sep. 2021.
- [15] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.
- [16] J. D. Rodriguez, A. Perez, and J. A. Lozano, "Sensitivity analysis of k-fold cross validation in prediction error estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, pp. 569–575, Dec. 2010.
- [17] Y. Wang, L. Li, and C. Dang, "Calibrating classification probabilities with shape-restricted polynomial regression," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 1813–1827, Jan. 2019.
- [18] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," *Association for Computing Machinery*, vol. 8, p. 625–632, Aug. 2005.

Robustness of Image Classification Using CNNs in Adverse Conditions

Theo Ingelstam and Johanna Skåntorp

Abstract—The usage of convolutional neural networks (CNNs) has revolutionized the field of computer vision. Though the algorithms used in image recognition have improved significantly in the past decade, they are still limited by the availability of training data. This paper aims to gain a better understanding of how limitations in the training data might affect the performance of the system. A robustness study was conducted. The study utilizes three different image datasets; pre-training CNN models on the ImageNet or CIFAR-10 datasets, and then training on the MADWeather dataset, whose main characteristic is containing images with differing levels of obscurity in front of the objects in the images. The MADWeather dataset is used in order to test how accurately a model can identify images that differ from its training dataset. The study shows that CNNs performance on one condition does not translate well to other conditions.

Sammanfattning—Bildklassificering med hjälp av datorer har revolutionerats genom introduktionen av CNNs. Och även om algoritmerna har förbättrats avsevärt, så är de fortsatt begränsade av tillgänglighet av data. Syftet med detta projekt är att få en bättre förståelse för hur begränsningar i träningsdata kan påverka prestandan för en modell. En studie genomförs för att avgöra hur robust en modell är mot att förutsättningarna, under vilka bilderna tas, förändras. Studien använder sig av tre olika dataset: ImageNet och CIFAR-10, för förträning av modellerna, samt MADWeather för vidare träning. MADWeather är speciellt framtaget med bilder där objekten är till olika grad grumlade. MADWeather datasetet används vidare för att avgöra hur bra en modell är på att klassificera bilder som tagits fram under omständigheter som avviker från träningsdatan. Studien visar att CNNs prestanda på en viss omständighet, inte kan generaliseras till andra omständigheter.

Index Terms—image recognition, convolutional neural network, CNN, adverse conditions, computer vision, MADWeather dataset

Supervisors: Saikat Chatterjee, Anubhab Ghosh

TRITA number: TRITA-EECS-EX-2022:177

I. INTRODUCTION

Machine learning systems used today are designed for limited tasks and fall in the category of narrow (or *weak*) AI. One such task is visual recognition, or image classification. Huge amounts of man- and computing power have been spent improving the technology, and not without impressive results: in 2015, He et al. designed an algorithm that surpassed humans on the ImageNet dataset, although noting that “this does not indicate that machine vision outperforms human vision on object recognition in general.” [1], suggesting we still have some ways to go. Krizhevsky et al. echoes this sentiment; writing that they “still have many orders of magnitude to go in order to match the infero temporal pathway of the

human visual system”, adding that “[a]ll of [their] experiments suggest that [their] results can be improved simply by waiting for faster GPUs and bigger datasets to become available.” [2]. While bigger datasets and faster GPUs (Graphics Processing Units) might certainly improve the performance of many contemporary image classifiers, this paper seeks to understand if there are some limitations inherent to CNNs that would prevent it from ever matching “the human visual system”.

The applications of visual recognition are broad. High level systems are integral to a lot of cutting edge technology, such as: autonomous vehicles and AI assisted medical diagnostics. In some areas computers excel, in large part due to the increased availability of large datasets. And while not all image classification systems need to exhibit human-like abilities (why limit a system built for species classification of birds to only know as many birds as the average human, or even the average ornithologist?) some do.

It is impossible to train *anything* on *everything* so in some applications, characteristics that can be considered inherent to the human brain: such as associative learning and lateral inhibition [3], might be needed. One such scenario is when the real life images the algorithm is asked to classify have been obtained under different conditions than the training dataset. While not always relevant - the image quality of X-rays or MRI scans, for example, are assumed to be consistent - in certain areas such as: autonomous vehicles and facial recognition, this eventuality is to be expected. In these cases, ensuring the robustness of the system, when tasked with classifying images taken under differing conditions, can be of utmost importance.

Research shows that CNNs perform worse when tested on ‘degraded images’ [4]. It is easy to fall into the trap of thinking that this is simply due to this being more *difficult*. It is true for humans, so why not for CNNs? Well, there is nothing that says that just because humans find a task difficult, the task will be complex for a computer [3]. And even though the two things might seem to coincide in this case we need to remember that the CNNs are initially trained on non-degraded images. It would therefore be more accurate to say that CNNs perform worse when asked to perform a task it was not trained for.

We propose that for CNNs, the task of classifying degraded images is not inherently a more difficult version of the task of classifying non-degraded images. If this was true, an algorithm that performs well on severely degraded images would automatically perform well, or maybe even better, on non-degraded images. In this paper we will test the robustness of an algorithm when faced with images captured under differing conditions from the training data. We will investigate how

models trained on images with different adversity conditions perform on images that somewhat differ from their training data, both with what humans would consider to be more adverse conditions and less. We design multiple different training scenarios, and compare the performance of models trained on them - using a dataset with a wide range of adverse conditions.

A major component in computer vision research and development is the availability of datasets. The need for large datasets when training and testing CNNs means that studies done on image classification under adverse conditions often have to rely on artificially degraded images, rather than using images captured under actual adverse conditions [4], [5], [6], and [7]¹. This due to a lack of large datasets that account for the factor 'adversity of condition under which the image was captured'. And while there are many obstacles to creating such a dataset, one issue is obvious; how to go about classifying 'adversity of condition'. When utilizing artificial tools to tune image quality one might refer to the number of transformations the image has undergone, and the severity of those transformations. When talking about real life adverse conditions such as: heavy rain, fog, and - as a consequence - light refraction, no such obvious classifiers exists.

In this paper we use the MAdWeather dataset, wherein the images were created specifically to provide images captured to simulate adverse weather conditions [8]. An in depth description of the MAd Weather dataset, can be found at [8]. Since, in practicality the images a model will have to classify will be gathered under actual adverse conditions we believe that it is optimal to do this type of testing using this type of dataset.

II. DATASET

A. Design

The MAdWeather dataset, designed by [8], has ten classes of fruits or vegetables: apples, bananas, carrots, lemons, onions, oranges, pears, peppers, potatoes, and tomatoes (referred to as objects). Each class consists of images of one or more of the object placed in a certain way, referred to here as a configuration. There are then seven images of each configuration with a varying number of plastic foil sheets between the camera and the objects to make the images more obscure. Each configuration has images where 0, 3, 6, 9, 12, 15, and finally 18 sheets of plastic foil have been placed in front of the object. We will refer to the number of sheets of plastic foils as the obscurity of an image, i.e. an image with fewer sheets is described as having a lower level of obscurity.

In total each class has 25 different configurations of the objects resulting in 1750 images in total. All images were taken with a neutral, white background. See Fig. 1 for examples of images from the dataset.

B. Different Datasets

For training and testing the dataset was divided into several subsets as follows:

¹Examples of artificial degradation include: addition of Gaussian noise, Gaussian blur, and random occlusions: as well as down-sampling, and variation of illumination.



Fig. 1. Examples of images from the MAdWeather dataset, each column contains images of a specific configuration of one of the objects and each row showing images with 0, 3, 6, 9, 12, 15 and 18 sheets in front of the object.

1) *Sheet Specific Datasets*: The data was divided so that each subset contained all images of all classes that were obscured by a specific number of plastic foil sheets. These subsets make it possible to train a model on images with a certain level of obscurity and testing the model on the other subsets to see how it performs under more or less obscure circumstances. When training with these subsets the subset used is further divided into training and validation sets, with a ratio of 60:40 respectively. The downside of this method is that we will be unable to test the models trained in this on an independent testing set depicting images with the same obscurity as those they were trained on.

2) *Interpolative Datasets*: The dataset is divided into training, testing and validation sets. The training set is composed of the images of seven configurations per object, but only the images that had 0 plastic sheets obscuring it and 18 plastic sheets. Thus, with two images from seven configurations of all ten objects the training set is composed of 140 images. The validation set was built in a similar way as the testing set but only choosing five configurations for each object. The remaining images are then used for the testing sets. By dividing the dataset in this way we will be able to test the models trained on these training and validation sets on independent testing sets of all levels of obscurity (the testing sets for 0 and 18 sheets of plastic being smaller since the training and validation images were removed from them).

3) *General Datasets*: The dataset is divided into training, testing and validation sets. The training set consists of all images from three configurations per class, resulting in 210 images. Similarly the validation set consists of all images from two other configurations per class, thus consisting of 140 images. The remaining 20 configurations from each class is

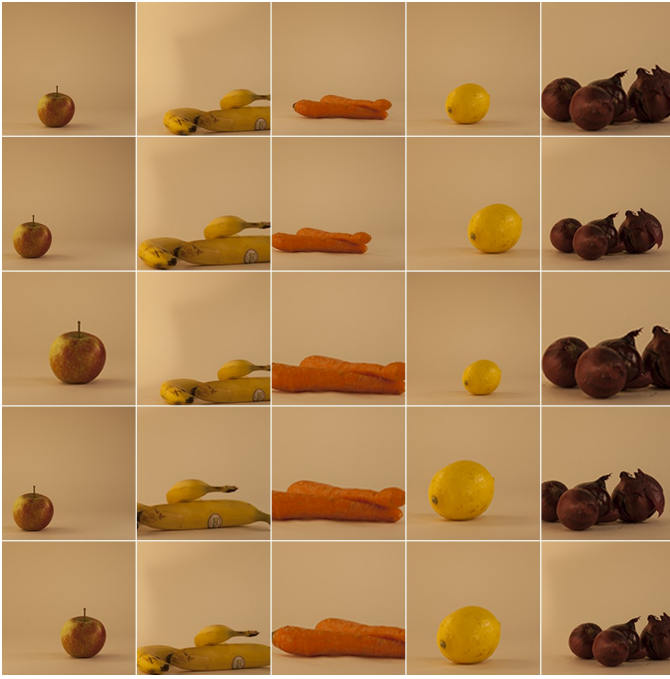


Fig. 2. Examples of the random transformations applied to the training images, the images here being configurations with 0 plastic sheets obscuring the objects.

then divided into testing sets, each set containing all remaining images with a specific number of sheets in front of the object, resulting in seven testing sets, each containing 200 images. The dataset is divided this way in order to have a model that has been trained on all levels of obscurity while still being able to test the model on independent testing sets.

C. Transforms on Training Data

Due to the limited amounts of images, the training dataset was artificially enlarged by altering each image through resizing and cropping into 224x224 pixels images. The dimensions of the images are made to fit with the expected input size of the CNN model used in this project. Each image in all different training datasets was altered nine times, saving the altered images as new images for training. This resulted in the training datasets being ten times as large as they would have been with just the original images. All resizing and cropping was made so that each object was still within the resulting image. See Fig. 2 for examples of the resulting images.

III. HUMAN TRIALS

In order to establish a baseline of human performance on the MAdWeather dataset, a study was conducted.

A. Method

Participants were shown 70 images in total, ten from each level of obscurity. The images were randomly chosen from the MAdWeather dataset. In order to improve the generalizability of the results, a unique image set was generated for each participant. The participants viewed one image at a time,

and were asked to identify the contents of the image as one of the ten categories of objects. The images were shown in decreasing level of obscurity, starting with 18 sheets of plastic and ending with 0.

B. Participation

There was a total of 84 respondents. The participants were friends and family members of the authors. Due to time limitations the test was hosted on a website [9]². This meant that, since respondents could take the test remotely, participation rate was high. A drawback was that there was slightly less oversight over testing conditions.

IV. TRAINING

The models used for testing were CNN models. CNNs, like other neural networks, are built up of layers of nodes that take several values for input data. In our case pixel values for the images we want classified. Where CNNs differ from other neural networks is that their middle layers consist of filters that, in theory, can be trained to detect patterns in adjacent data points. These layers are called convolutional layers. By connecting several such layers the model can feasibly go from detecting for example curved or straight lines to detect complicated shapes such as facial features. After the last convolutional layer the detected patterns positions are weighted so as to represent a numerical value. These numerical values are then used in one or more linear layer to make a prediction about which of a predetermined set of classes the image belongs to. When training a CNN the filters and weights are adjusted in order to make the model's predictions be as correct as possible for the training data, these adjustable filters and weights are called features [10]. When training a model that has already been trained previously on a different dataset one can choose to either adjust the whole model, including filters, or only train the final linear layer. Training only the final layer will go faster but will only yield accurate predictions if the pre-trained filters are relevant for the new dataset.

Due to limited computing power a CNN model was needed that had relatively few trainable features, while still having a high accuracy on a benchmark dataset, in our case ImageNet [11]. The SqueezeNet model, developed by [12], fulfills these criteria well. It has an accuracy comparable to top of the line CNN models, while having only a fraction of the features. An overview of the SqueezeNet model structure can be seen in Fig. 3.

A. Model Structures

We use four different model structures for our CNN.

1) \mathcal{J}_1 : The SqueezeNet model is loaded with features pretrained on the ImageNet dataset. The final layer is then reshaped to only have ten output classes. Finally the all layers of the model are frozen except for the final layer, so that training will only be done on the final layer of the model.

²This was also done in part because of COVID concerns.

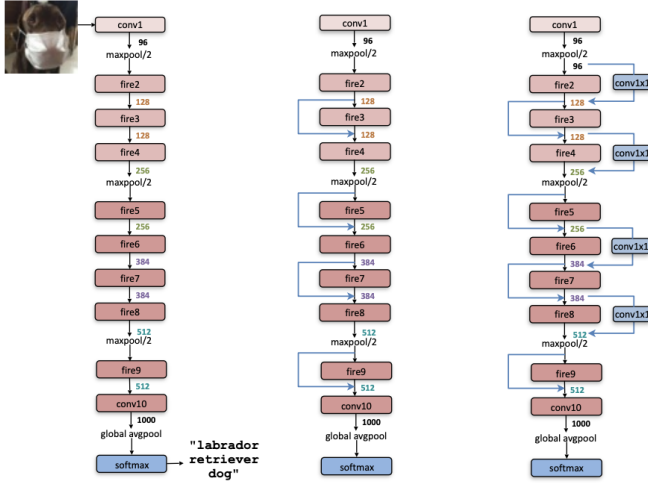


Fig. 3. Overview of SqueezeNet’s model structure. The pooling layers enable better performance on smaller datasets, such as MAdWeather. Further details about the different steps of the model can be found in [12].



Fig. 4. Examples of images in the CIFAR-10 dataset after being resized from 32x32 pixels to 224x224 pixels.

2) \mathcal{I}_2 : The SqueezeNet model is loaded in the same way as in \mathcal{I}_1 . As opposed to \mathcal{I}_1 however, no layers of the model are frozen during future training. This means that all features are being trained, including all convolutional layers.

3) \mathcal{E}_1 : The SqueezeNet model is loaded with randomized features. The final layer is then reshaped to only have ten output classes. The model is then trained on the CIFAR-10 dataset which, similar to the MAdWeather dataset, only has ten classes. Since SqueezeNet recommends a minimum image size of 224x224 pixels and CIFAR-10 consists of 32x32 pixels images the dataset was upsampled in order to train the model, Fig. 4 shows how the images look after upsampling. After

training, the model had a 90% Top-1 performance on the testing images in the CIFAR-10 dataset, meaning the number of images that were correctly classified. Finally all layers of the model are frozen except for the final layer, and the final layer is retrained, similarly to \mathcal{I}_1 .

4) \mathcal{E}_2 : The SqueezeNet model is loaded and trained on CIFAR-10 in the same way as in \mathcal{E}_1 , but much like \mathcal{I}_2 no layers of the model are frozen when the model is later trained on the MAdWeather dataset.

Our motivation for training with several different models is to ensure that when we later analyze the results we will be able to analyze the model that has the best accuracy, and therefore is the model best suited for the MAdWeather dataset. SqueezeNet already has a version that has been thoroughly trained on ImageNet, but ImageNet consists of one thousand different classes [11]. We therefore became interested in seeing if models that were pretrained on a dataset that, like the MAdWeather dataset, contained only ten classes yielded better accuracy after training. CIFAR-10 fulfills the criteria of consisting of ten classes. Note that the classes in the CIFAR-10 dataset are not the same as the objects in the MAdWeather dataset.

Finally we wanted to compare models where only the final layer was trained on the MAdWeather dataset against models where all layers of the model were retrained. This was done in order to verify if the filters used for classifying ImageNet or CIFAR-10 images would be well suited for identifying MAdWeather images, or if training the convolutional layers resulted in notably higher accuracy for identifying the testing images. ImageNet and CIFAR-10 are publicly available datasets often used for image recognition and were perceived as applicable for the training done in this project.

B. Training Scenarios

The above model structures are all trained in three different ways, using datasets defined under subsection II-B Different Datasets.

1) *Sheet Specific Training, \mathcal{S}* : The models are specifically trained on images from one level of obscurity. The levels chosen are 0 (\mathcal{S}_0), 9 (\mathcal{S}_9), and 15 (\mathcal{S}_{15}) plastic foil sheets. The training dataset consists of 150 images, all from the same level of obscurity, as described in more detail under subsection II-B1 Sheet Specific Datasets. This method is used to ascertain how well models trained under specific circumstances generalize on testing sets that differ more or less from the training circumstances.

2) *Interpolation Training, \mathcal{I}* : The models are trained on both the lowest and highest level of obscurity, as described under subsection II-B2 Interpolative Datasets, with the training set consisting of a total of 140 images split evenly between the two. This is done to study how well models can interpolate, when tested on obscurity levels that lie ‘in between’ the training cases.

3) *General training, \mathcal{G}* : The models are trained, simultaneously, on all levels of obscurity. The training set consists of 30 images per level of obscurity, for a total of

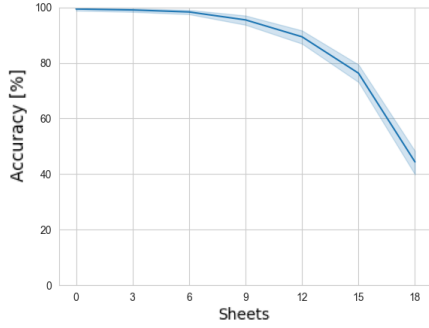


Fig. 5. Human performance \mathcal{H} on the MAdWeather dataset. The mean value for the 84 respondents are plotted with a 95% confidence interval.

210 images, as described under subsection II-B3 General Datasets. This will give us a good estimate of how well a model can perform when trained in all available conditions.

The goal is for the training scenarios to be, to the maximum extent, comparable. The problem that arises is this: if all scenarios are provided the same number of images per level of obscurity to train on, they will be trained on a *very* different number of images in total. Then again, if all scenarios are provided the same number of images in total to train on, they will be trained on a *very* different number of images per level. For our purposes, we have decided to keep the total size of the training dataset comparable, rather than for each level. There are still some minor discrepancies between the sizes of the training datasets for the different scenarios, the reasons for which are further expounded upon in subsection II-B Different Datasets.

V. RESULTS

A. Human Performance

We present the mean value for human performance with a 95% confidence interval in Fig. 5. As expected human performance goes down as the level of obscurity goes up. There is a risk that human performance on the highest level of obscurity was negatively impacted by the fact that the study was conducted remotely and these were the first images they were shown. Any initial confusion as to the design of the study would therefore disproportionately affect this part of the responses.

B. SqueezeNet Performance

The testing results for all training scenarios are shown, for the four model structures, in Fig. 6, using human performance as comparison. The results are reported as Top-1 accuracy.

Note that for the training scenarios \mathcal{S}_i ($i = 0, 9, 15$) all images with i number of plastic sheets were used in training. Therefore, the testing data for level i on \mathcal{S}_i is part of the training data, and will not be included in the discussion below. These data points are marked with a red cross in Fig. 6.

For ease of comparison between the different training scenarios we also include Fig. 7. We plot the top-1 accuracy for

\mathcal{I} and \mathcal{G} , as well as the performance of \mathcal{S}_i on neighboring levels of obscurity (i.e. for \mathcal{S}_9 we include its performance on 6 and 12 sheets of plastic).

C. Model Structure

1) *Frozen layers* (\mathcal{J}_1 and \mathcal{C}_1): We can see that when the model is pre-trained on ImageNet, its features are better suited for the MAdWeather dataset. Opening up all layers for training improves performance on almost all training scenarios³, under all testing conditions, meaning that the models where only the final layer was trained performs worse. This is to be expected. Only training the final layer serves to give us a baseline understanding of the models before we fully train all layers.

2) *ImageNet* (\mathcal{J}) and *CIFAR-10* (\mathcal{C}): Results show that models pretrained on ImageNet perform better than models pretrained on CIFAR-10. There can be several reasons for this. One reason could be that due to limited processing power we were unable to pre-train the models on CIFAR-10 to the same degree as the pre-training that had been done with ImageNet by default. Also, since the CIFAR-10 dataset consists of lower-resolution images, upsampling was performed, resulting in blurry training images, as can be seen in Fig. 4.

All models show better overall performance on lower levels of obscurity, although the models trained on CIFAR-10 does not show as strong a correlation between the level of obscurity and performance as those models trained on ImageNet. And exceptions can be found. For example \mathcal{S}_9 in \mathcal{C}_1 perform better on 12, 15, and 18 sheets, than on 0, 3, and 6 sheets, as seen in Fig. 6, and \mathcal{G} in \mathcal{C}_2 perform better on 3-12 sheets, than on 0 sheets.

We observe that pre-training the model on ImageNet provides features better suited for the MAdWeather dataset, but that a model pre-trained on CIFAR-10 shows less bias to obscure images.

D. Training Scenarios

1) Sheet Specific Training \mathcal{S} :

- For sheet specific training \mathcal{S} , opening up all layers have the greatest impact on 'neighboring' levels of obscurity, and little to none effect on obscurity levels 'far away'.
- Both \mathcal{S}_9 and \mathcal{S}_{15} perform better on the neighbor with lower obscurity, than the neighbor with higher obscurity.

2) Interpolation Training \mathcal{I} :

- Interpolation training (\mathcal{I}) performs similarly to \mathcal{S}_0 on 3 sheets of plastic and similarly to \mathcal{S}_{15} on 12 sheets of plastic.
- \mathcal{S}_9 perform significantly better on 6 sheets of plastic than \mathcal{I} , and \mathcal{I} perform significantly better than \mathcal{S}_9 on 12 sheets of plastic.
- Performs much better on 9 sheets of plastic, than either \mathcal{S}_0 och \mathcal{S}_{15} .

³With the exception that \mathcal{S}_9 correctly classifies 26.4% of images using \mathcal{C}_1 , which drops to 12.1% for \mathcal{C}_2 .

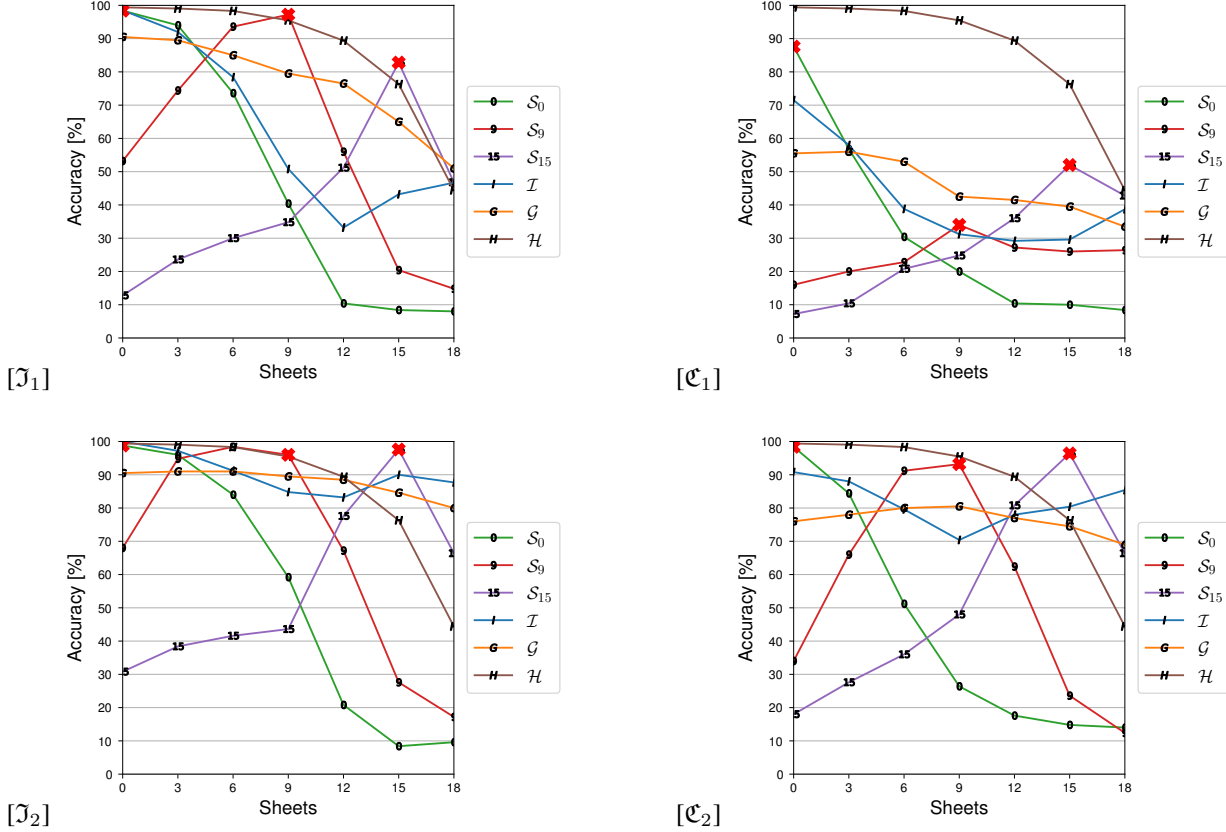


Fig. 6. Top-1 performance of all training scenarios, on all four model structures, as well as mean value of human performance (\mathcal{H}) on the MAdWeather dataset.

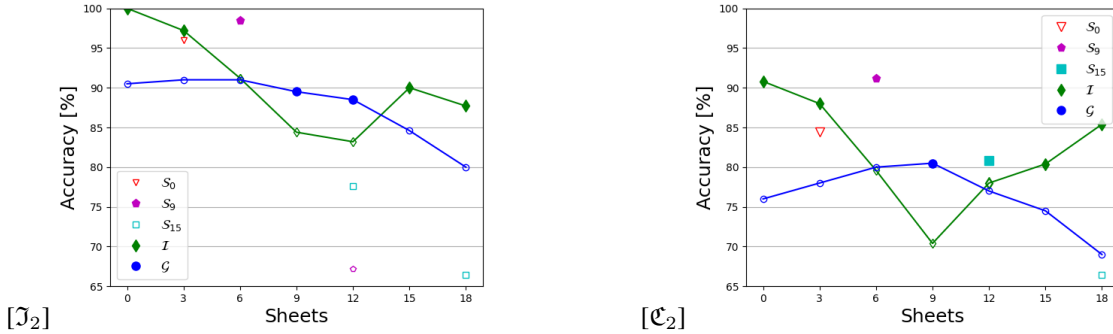


Fig. 7. Performance of \mathcal{G} and \mathcal{I} , as well as the performance of neighboring levels for all \mathcal{S} . The marker for the highest performing scenario for each level is fully colored.

3) General training \mathcal{G} :

- In \mathcal{C}_2 the general training \mathcal{G} performs better on mid-levels of obscurity than on high and low levels.
- Overall best performance, but rarely performs best on any single level. \mathcal{G} has top performance in three test cases, two of which are 9 sheets of plastic. This is a level we expect all other training scenarios to perform badly at.

Due to time restrictions we had to limit the number of training scenarios. This becomes problematic in two ways:

- Since all images with i number of plastic sheets were used in training \mathcal{S}_i , we can only compare neighboring

levels of obscurity. Unfortunately \mathcal{S}_0 only has one neighbor (with a higher level of obscurity), whereas \mathcal{S}_9 and \mathcal{S}_{15} has two (one lower and one higher).

- When comparing training scenario \mathcal{I} , which was trained on 0 and 18 sheets of plastic, to the sheet specific training \mathcal{S} , it would have been preferable to compare with \mathcal{S}_0 and \mathcal{S}_{18} , rather than \mathcal{S}_0 and \mathcal{S}_{15} .

It would have been preferable to also have done training and testing on \mathcal{S}_3 and \mathcal{S}_{18} .

VI. DISCUSSION

A. Limitations of the Study

The two major limitations are time and the relatively small size of the MAdWeather dataset. Throughout the text we have mentioned how these limitations might affect our results. Additionally the design of the MAdWeather dataset may also hold some implications. [5] is a study on the effect of performance degradation, due to parameters such as: illumination, noise, motion blur, and resolution. The study concerns facial recognition and show that if the relative pose difference between test and reference image is small, the effect of such degradation grows. The images in the MAdWeather dataset are relatively pose invariant, which might affect our results.

We also want to reiterate that the datasets for our training scenarios are constructed on the basis of 'instead of', as opposed to 'in addition to'. As an example: \mathcal{G} is trained on 30 images from each level of obscurity, for a total of 210 images, while \mathcal{S}_0 is trained on 150 images with 0 sheets of plastic. This is further explained under subsection IV-B Training Scenarios.

B. Complexity vs. Difficulty

As noted; all models show better overall performance on the lower levels of obscurity. It is hard to determine whether this is because it is a less complex task, or if it is the result of the pre-training done on non-degraded images. The CIFAR-10 dataset consists of lower quality images (see Fig. 4) than ImageNet, and the models pre-trained on CIFAR-10 are less inclined to favor less obscure images (in some cases performing better on more obscure levels). This could indicate that classifying more obscure images is not an inherently more complex task. Again, with a larger dataset we could provide a better answer. With more data, the pre-training would not have to be done on non-degraded images, which would provide more generalizable results.

C. Training Scenarios

It is clear that if the model has high performance on high levels of obscurity, it does not automatically mean that it has a high performance on low levels of obscurity. Classifying images that a human would think of as 'easy' is not simply a less complex version of classifying images that a human would think of as 'complex'. Even when the model has high performance on both low and high levels of obscurity (\mathcal{I}), that does not imply it has high performance on middling levels. If the model were to interpolate well we would expect to see similar accuracy on all obscurity levels for \mathcal{I} . Since this is not the case we can say that the models used do not interpolate reliably. In order to get better results on images with mid-level obscurity the model needs to be trained on images that have a similar obscurity, such as \mathcal{S}_9 and \mathcal{G} .

[4] suggest that a better approach to creating image classifiers that are robust to adverse conditions might be to "[...] design models specialized to each kind of degradation, separately." Seeing how much better \mathcal{S}_9 performs, on 6 sheets of plastic, than \mathcal{I} , our results support this.

D. Artificial Degradation vs. Adverse Conditions

In this study we did not focus on how well one can replicate actual adverse conditions, using artificial degradation. In [6], they show, using artificially degraded data, that some degradation to training data quality might improve the ability of the algorithm to identify less degraded images. This is not something we have necessarily been able to replicate using our dataset. This might be because the 'step' between each level of obscurity is large. It would be interesting to construct a similar dataset to ours, using only artificially degraded images, and replicate the testing done here. We believe the MAdWeather dataset would be suitable to this due to it being relatively pose invariant.

VII. CONCLUSION

We outline how the MAdWeather dataset can be used to research how well image classification algorithms can be trained on actual adverse weather conditions, using artificially degraded images. If such research would show that artificial degradation is a poor substitute for 'bad weather', it could have big implications, since most research relating to this is done using artificially degraded images. We suggest how to expand upon the work done here, given more time and, most importantly, more data.

We show just how important it is to be rigorous when testing performance of image classifiers in adverse conditions. Just because it performs well on severely degraded images, that says very little of its performance on other, even 'easier' levels. We can not just test on the 'easiest' and 'hardest' images and assume a similar performance across the board. Testing needs to be done on all obscurity levels in order to ensure quality.

ACKNOWLEDGMENT

We would like to thank our faculty supervisor Saikat Chatterjee. We also extend our gratitude to Anubhab Ghosh, for taking the time and talking us through any and all problems we brought his way.

A big thank you to the authors of [8] for their work creating the MAdWeather dataset, without which this thesis work would not exist.

We also thank Fredrik Burmester, whose help with the website [9] for the human trials increased our response rate; from an initial goal of 30 participants to the actual 84.

REFERENCES

- [1] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, vol. 2015. IEEE, 2015, pp. 1026–1034.
- [2] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [3] J. E. H. Korteling, G. van de Boer-Visschedijk, R. Blankendaal, R. Boonekamp, and A. Eikelboom, "Human- versus artificial intelligence," *Frontiers in artificial intelligence*, vol. 4, pp. 622 364–622 364, 2021.
- [4] S. Karahan, M. K. Yildirim, K. Kirtac, F. S. Rende, G. Butun, and H. K. Ekenel, "How image degradations affect deep CNN-based face recognition?" in *2016 International Conference of the Biometrics Special Interest Group (BIOSIG)*. IEEE, Sep. 2016.

- [5] A. Dutta, R. Veldhuis, and L. Spreeuwens, "The impact of image quality on the performance of face recognition," in *33rd Symposium on Information Theory in the Benelux and the 2nd Joint WIC/IEEE Symposium on Information Theory and Signal Processing in the Benelux*. Netherlands: Werkgemeenschap voor Informatie- en Communicatietheorie (WIC), May 2012, pp. 141–148.
- [6] D. Liu, B. Cheng, Z. Wang, H. Zhang, and T. S. Huang, "Enhance visual recognition under adverse conditions via deep networks," *IEEE Transactions on Image Processing*, vol. 28, no. 9, pp. 4401–4412, 2019.
- [7] M. A. Butt, A. M. Khattak, S. Shafique, B. Hayat, S. Abid, K.-I. Kim, M. W. Ayub, A. Sajid, and A. Adnan, "Convolutional neural network based vehicle classification in adverse illuminous conditions for intelligent transportation systems," *Complexity (New York, N.Y.)*, vol. 2021, pp. 1–11, Feb. 2021.
- [8] "Madweather: Object recognition under a mock model of adverse weather conditions and study of data-limited machine learning against human," report, KTH, Stockholm, Sweden, 2019.
- [9] T. Ingelstam, J. Skåntorp. (2022) Human performance test. Stockholm, Sweden. [Online]. Available: <https://fruitsbachelor.fdrive.se/>
- [10] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6.
- [11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
- [12] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size," 2016.

Pre-analysis of Nanopore Data for DNA Base Calling

Milad Javadi and Yun Luk Liu

Abstract—Nanopore sequencing is a relatively new DNA sequencing method which measures the current over a nanopore in a membrane as each nucleotide of the DNA passes through the nanopore. From the resulting current signal it is possible to determine the sequence of nucleotides in the DNA by using a base caller. The goal of this project was to create a machine learning model which could estimate the accuracy rate (identity score) of the sequenced DNA using the electric current signal and other data available through nanopore sequencing. The dataset that the machine learning models were trained on were samples from *E. coli* bacteria that had been sequenced through nanopore sequencing. In this project a linear regression model was created as well as several neural networks. The best performing model was a neural network which had a mean square error (MSE) of $6.12 \cdot 10^{-4}$, compared to a variance in the dataset of $2.11 \cdot 10^{-3}$. The low MSE indicates that the model can effectively predict identity scores.

Sammanfattning—Nanopore sequencing är en relativt ny DNA-sekvenseringsmetod som mäter strömmen över en nanoskopisk por i ett membran samtidigt som varje DNA-nukleotid passerar genom poren. Från den resulterande elektriska signalen så är det möjligt att bestämma sekvensen av nukleotider i DNA:t genom att använda en base caller. Målet med det här projektet var att skapa en maskininlärningsmodell som kunde bestämma graden av noggrannhet av det sekvenserade DNA:t genom att använda den elektriska strömsignalen och andra typer av data tillgängliga av Nanopore sequencing. Datamängden som maskininlärningsmodellerna använde för träning bestod av samples från en *E. coli* bakterie som sekvenserats med nanopore sequencing. I det här projektet har en linjär regressionsmodell skapats samt flera olika neurala nätverk. Den bäst presterande modellen var ett neuralt nätverk, som hade en minstkvadratfel (MSE) på $6.12 \cdot 10^{-4}$, jämfört med datamängdens varians på $2.11 \cdot 10^{-3}$. Det låga MSE-värdet visar på att modellen effektivt kan skatta noggrannhetsgraden av den avlästa DNA-sekvensen.

Index Terms—Nanopore sequencing, DNA sequencing, bioinformatics, machine learning, neural networks, linear regression, supervised learning

Supervisors: Joakim Jaldén, Javier Kipen, Xuechun Xu

TRITA number: TRITA-EECS-EX-2022:178

I. INTRODUCTION

This project builds on the foundation of a new technique for sequencing DNA called nanopore sequencing. Nanopore sequencing utilizes a membrane containing a nanopore (essentially a nanoscopic hole) to isolate two phases of a solution containing some concentration of electrolytes and adenosine triphosphate (ATP). An ionic current is induced across the membrane. A DNA molecule is then first split into two RNA molecules, and then passed through the nanopore, allowing it to act as a resistor, and modulate the current. This current is measured and recorded as a signal over time [1].

If unregulated, the RNA will pass through the nanopore at a speed too fast for the current sensor to make a reliable measurement. Therefore, a so called motor protein is integrated inside the nanopore. The motor protein acts as a stepper motor, restricting the movement of the RNA molecule. It consumes a single ATP molecule from the electrolyte solution at random intervals, and in turn lets the RNA advance the equivalent distance taken up by one of its base pairs [2].

The different primary nucleobases that make up the DNA molecule differ in conductive ability. While the purine bases adenine (A) and guanine (G) both have a relatively low conductance, the pyrimidine bases cytosine (C) and thymine (T) have higher conductances (see Fig. 1). As such, the current across the membrane will vary in magnitude depending on what bases are being processed by the nanopore (see Fig. 2) [2]. For the purposes of this report, all currents in the presented data have been z -score normalized. That is, they have been transformed such that the mean $\mu = 0$ and variance $\sigma^2 = 1$ for each recording.

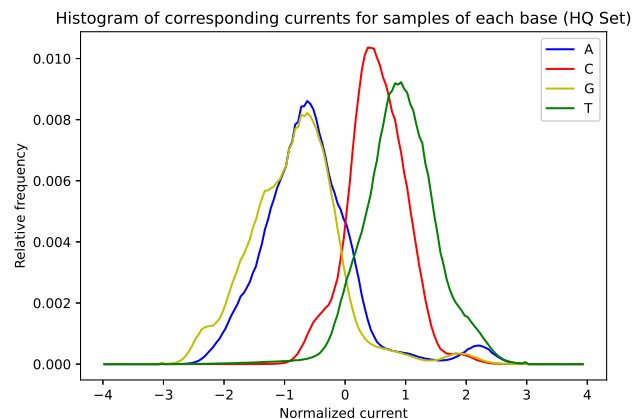


Fig. 1. Histogram of current levels that correspond to each base, extracted from the high quality dataset used in the project.

From the recorded current signal, the sequence of nucleobases can be estimated using an algorithm known as a base caller. For the purposes of this project, the base caller is a machine learning-based algorithm that has been trained on current recordings of DNA segments from various organisms, along with reference sequences of those same segments, derived using conventional sequencing methods [1].

The process of sequencing DNA in this manner involves several processes that are stochastic in nature, which introduces variance into the system. The biggest of these factors is the intervals at which the motor protein advances the DNA

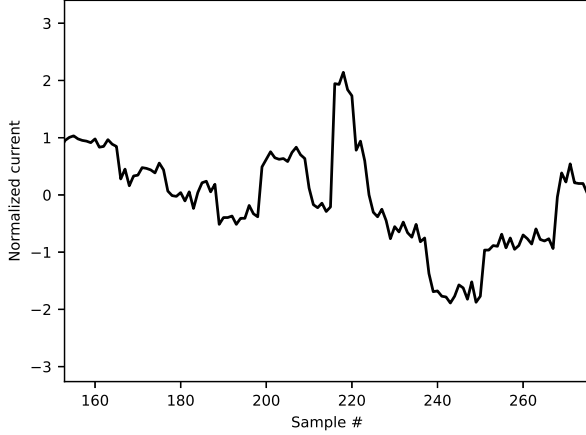


Fig. 2. Normalized current level of a recorded signal over time, as different bases pass through the nanopore. Example taken from a single recording from the high quality dataset used in the project.

molecule. As such, it is currently nearly impossible to derive a fault-free sequence using nanopore sequencing. According to [3], the single-read accuracy for nanopore sequencing was, in 2020, around 99%, which is rather close to, but still not perfect. Since even tiny deviations in genetic composition can have drastic implications on an organism's features, this becomes a serious problem when studying DNA samples which have not previously been sequenced using conventional methods [4].

One way to evaluate the performance of a nanopore sequencing system is by comparing the sequence it produces with a reference sequence. The reference sequence is derived from a similar DNA sample, but using conventional sequencing methods. The identity score s of a sequence is a measurement of its accuracy, relating to the total number of matching nucleotides n_{match} , substitution errors E_{sub} and insertion/deletion errors E_{indel} between it and its corresponding reference sequence.

$$s = \frac{n_{match}}{n_{match} + E_{sub} + E_{indel}} \quad (1)$$

Identity scores are stored as floating-point values ranging between 0 and 1, where 1 indicates a perfectly matching sequence, and 0 indicates a sequence with no matching bases [5]. A sequence will, in reality, never achieve an identity score of 0. Due to how the identity score is calculated, even two completely non-related sequences will show some level of matching, and a particularly low-quality sequence will reach scores upwards of 0.6 to 0.7 [6].

To improve sequence accuracy, several so-called post-sequencing techniques are used. Among these are consensus calling and consensus polishing, which involve combining parts of several sequences into a single sequence of higher quality. As described in [2], these techniques benefit from starting with sequences that are already relatively high quality to begin with.

Given enough development, nanopore sequencing will hope-

fully become sophisticated enough to act as a faster, more accessible alternative to conventional sequencing methods. It already has multiple advantages over conventional methods. Besides faster sequencing times, it is also able to read longer DNA segments, and the devices used take up significantly less space [7].

The goal of this project was to create an additional estimator. One that receives a data entry consisting either only of a current signal recording (i.e. an end-to-end model), or the signal as well as some computed features from its base calling. The estimator would then output a predicted identity score for the derived sequence. The use for such an estimator would be in tools made for aligning a nanopore-derived sequence with either another nanopore-derived sequence, or a reference sequence (see [8]). These tools could make use of an estimated identity score as an indicator of when two sequences are properly aligned.

An end-to-end model would provide a few advantages over a feature-based one, in that it would allow for sequence quality evaluation even before base calling. This makes it possible to omit low-quality signals directly after reading, and only sequence high-quality signals. It would also be compatible with any base caller, even if it does not output the same features as the ones used in this project.

II. METHOD

A dataset consisting of recorded current signals and their corresponding identity scores was provided. The signals were of varying quality, and were divided into two subsets: a high quality dataset, with a median identity score of 95%, and a low quality dataset with a median identity score of 90%. The higher quality dataset held 4697 data entries, while the lower quality dataset held 4098 data entries. All the data points represent recorded current signals of an *Escherichia coli* (E. coli) genome passing through a nanopore. Along with the current signals, several additional features of the data entries were provided and they are as following:

- *Identity*: The identity score of the derived sequence
- For each sample in the recording:
 - *Base*: Which primary base (A, C, G or T) the sample corresponds to
 - *Alignment*: The ordinal number of the base that the sample corresponds to (for example, if the alignment of a sample is 12, it means that the sample corresponds to the 12th base in the sequence, starting from 0)
 - *Alignment probability*: A measurement of how confident the base caller is that the sample in fact corresponds to its assigned base

A. Features of interest

From the provided features, several others were extrapolated and used as input data when training. A total of twelve different features were computed and evaluated:

- The mean, and standard deviation of the time (in samples) each base is recorded for¹
- The mean, and standard deviation of the alignment probability of each sample in the recording
- The means, and standard deviations of the current levels that correspond to each base in the recording

In order to identify potential features that could be useful for the machine learning models, scripts were created that could visualize the data both for single reads of the dataset, as well as for all reads together. All programming work was done in Python and all plots were created using the Matplotlib library.

One of the scripts that were created was an inspector program. It would inspect the data for specific reads, showing the duration distribution of the nucleobases passing the nanopore, the distribution of nucleobases for the given read, and the distribution of the normalized signal. This program was used to try to find a correlation amongst the data for reads with high identity scores. By studying the plots for similarities, any pattern that could be observed would be a feature of interest for the machine learning models. Fig. 1 shows an example of the distribution of the bases that were studied to find possible correlation between features. In the figure it is possible to see the overlap and the variances of the current for each base.

The inspector program described above was used first and foremost to study reads with the highest identity scores as well as reads with the lowest identity scores. This was done in order to investigate if there are any patterns emerging for those reads. If patterns could be found for reads with high identity scores that could be a possible correlation which could be fed as an input to a machine learning model. If there would be any patterns for lowest identity score reads, those could be used to find features that were negatively correlated to the identity score, which would also be useful for machine learning models.

Two features of interest to check for their correlation with the identity score were the mean duration for the nucleobases to pass the nanopore and the standard deviation of the durations. It is because it was discovered, with the inspector scripts, that samples with very high durations also tended to have lower identity scores. Another feature that was of interest was the mean alignment probability. To study the correlation with the identity score, the mean alignment probability and the identity score were plotted on a 2D-histogram which is shown in Fig. 4. From the 2D-histogram it is possible to see a potential correlation with the identity score. The shape of the figure shows that there tends to be a higher identity score with a higher mean alignment probability.

At last, before starting to work on the models, a correlation matrix was created to compute and visualize the correlation between several features. Any potential feature of interest identified from the scripts above were fed into a correlation matrix to find their correlation with the identity score. From the correlation matrix that was produced, shown in Fig. 3, the highest correlated features were then chosen to be tested as inputs to the machine learning models. The correlation

matrix also shows how correlated the input features are with one another. This becomes important for feature selection, as selecting two features which are highly correlated with one another (for example mean alignment probability and std. of alignment probability) likely will not be as beneficial as selecting two uncorrelated features. The reason for this is that two correlated features will provide mostly the same information when evaluated, so evaluating them both becomes redundant.

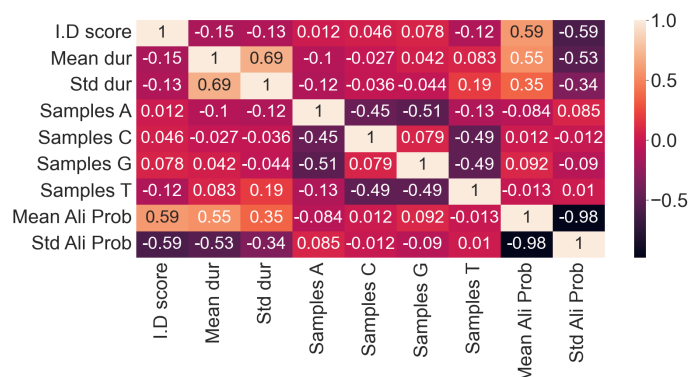


Fig. 3. Correlation matrix of the combined datasets of both the lower quality dataset and the higher quality dataset.

For the neural network models, feature selection was made simple by constructing a decision tree from the dataset using the Scikit-learn package in Python, and extracting feature importances from that. The features with the highest relative feature importance are expected to provide the largest information gain when evaluated, and are therefore favorable to use in training. The decision tree was only used in conjunction with the correlation matrix. This is to avoid selecting redundant features which are highly correlated with one another. Feature sets obtained from both the correlation matrix alone, and the decision tree along with the correlation matrix were tested with the neural networks.

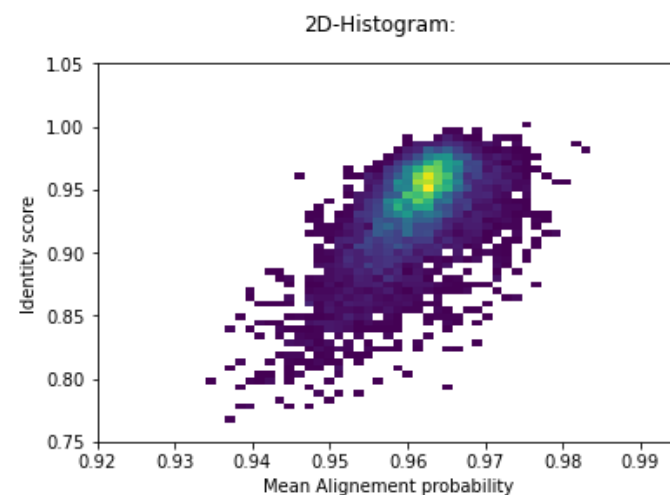


Fig. 4. 2D histogram which shows distribution of samples in regards to their identity score versus mean alignment probability.

¹This is calculated by dividing the total length of the signal by the number of different bases in the sequence.

B. Models

For this project, several different kinds of estimators were tested. Most of the estimator types have adjustable parameters, many of which were tweaked, both systematically and non-systematically in order to improve performance. The models that represent each estimator type in the results are the single best performing ones of each type.

1) *Constant estimator*: The constant estimator simply estimated the identity score of any given reading to be equal to the mean identity score of the entire dataset. This was done to establish a baseline against which the other models could be compared. Its mean square error (MSE) will by definition be equal to the variance of all identity scores in the dataset [9].

2) *Linear regressor*: For the linear regressors, the average of the durations for each base in a reading was plotted against the reading's identity score, in an attempt to derive a linear relationship. The linear regression model used an 80/20 train/test-split which means that 80% of the data was used for training the model, while 20% of the data was reserved for verifying how well the model perform on the test data. The model was implemented using Scikit-learn package in python. The linear regressor works by finding a linear approximation model which minimizes the distance between the data points in the dataset and the linear model.

3) *General neural network*: General neural networks are essentially graphs built from several layers of nodes, called neurons. There are three primary kinds of neurons:

- *Input neurons*, which output an input value to several other neurons
- *Output neurons*, that take in a number of values as input, and then weights them, adds them together, and finally adjusts the sum with a bias before compressing the value using an *activation function*, such as the sigmoid function (see eq. (2)). The resulting value is part of the neural network's output
- *Hidden layer neurons*, that, like an output neuron, takes in and processes input values, but outputs them to either output neurons or other hidden layer neurons (see Fig. 5)

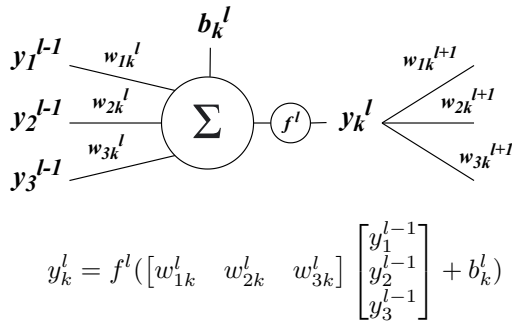


Fig. 5. Example of a hidden layer neuron k in hidden layer l , with output y_k^l , bias b_k^l , and activation function f^l .

A typical neural network consists of a layer of input neurons, connected to one or more layers of hidden layer neurons, connected finally to a layer of output neurons (see Fig. 6).

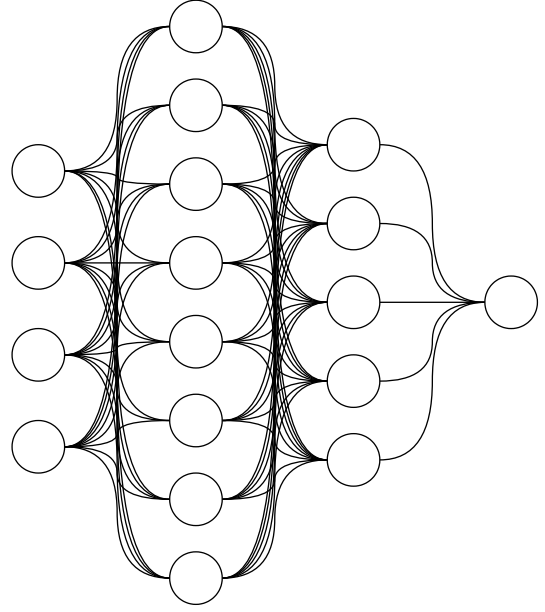


Fig. 6. Example of a neural network with 4 input neurons, (8, 5) hidden layer neurons, and 1 output neuron.

The neural network is initialized with random values for all its weights and biases. After it receives a set of inputs, an output is calculated, and then compared against a reference output. Depending on the difference between the reference output and the calculated output, the weights and biases of the network are adjusted using a *loss function*, often together with an *optimizer*, so that it will in the future produce an output that is closer to the reference. This process is referred to as *training* the network. After having trained on every data entry in the set, the network will have completed one *epoch* of training.

$$S(x) = \frac{1}{1 + e^{-x}} \quad (2)$$

For this project, the Python package Keras was used to construct the neural networks, with an input layer consisting of the features evaluated, two hidden layers, and a single output neuron which corresponds to the predicted identity score.

Input features were selected using the correlation matrix, as well as the decision tree. In order to make training more efficient, all input data was z -score normalized along each feature, as advised in [10].

All hidden layer neurons used the rectified linear function (ReLU) as activation function, (see eq. (3)).

$$R(x) = \max(0, x) \quad (3)$$

To decide on the appropriate number of neurons in each hidden layer, every possible model consisting of 20—50 neurons in the first hidden layer, and 10—40 neurons in the second hidden layer was sequentially evaluated. These bounds were originally determined through trial and error during initial test runs of the neural network. During these runs, a single-hidden layer neural network was first tested with largely varying numbers

of hidden layer neurons. Using the best performing single-hidden layer network as a basis, a second hidden layer was then added on. The neuron count of the second hidden layer was similarly changed between runs in a heuristic manner. This gave a rough idea of what range of neuron counts would produce the best performing networks.

MSE was used as the loss function for these networks (see eq. (4)), with *stochastic gradient descent* (SGD) as the optimizer. In Keras, SGD has an adjustable *learning rate* parameter, which determines how much the loss function affects the weights and biases after each round of training. This was not explicitly evaluated during the project, but it was periodically adjusted between trials to see how it would affect the performance of the networks. A technique known as *early stopping* was also used. It involves stopping the training if the model has not improved within the last five epochs.

The general neural networks used an 85/15 train/test-split.

4) *Convolutional neural network*: Convolutional neural networks (CNNs) work much in the same way as general neural networks. The key difference is that between the input layer and hidden layers there are several layer of convolution. The technical aspects of what this layer does and looks like is beyond the scope of this paper, but is explained in detail in [11]. The main takeaway is that features in a CNN are not represented individually, but rather in groups of adjacent features. This means that if the set of input features used is positionally dependent, e.g. a current signal, the CNN may be able to identify local patterns that affect the output [2]. This makes CNNs a notable candidate for end-to-end learning. Besides the benefits mentioned in section I, end-to-end neural networks generally outperform feature-based neural networks [6].

For this project, Keras was again used to construct the CNNs, with an input layer consisting of the entire 4096-sample signal, two convolutional layers with a size 3 kernel (see [11]), two hidden layers consisting of 64 neurons each, and a single output neuron.

Like the general neural networks, the convolutional neural networks used MSE and SGD as their loss function and optimizer, respectively, as well as an 85/15 train/test-split.

C. Performance evaluation

The performance of each model is evaluated with regards to its MSE for a particular training session over the entire dataset (of size N). That is, how far its estimation \hat{s}_i is, on average, from the actual identity score s_i of the i^{th} sequence in the dataset.

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (\hat{s}_i - s_i)^2 \quad (4)$$

All identity scores used in training were z -score normalized. This was done to allow the models to be evaluated independently from the spread of the training data. A dataset with higher variance would in this case be harder to make predictions on than one with a lower variance. As such, the results will be more tangible if the models are instead represented by how they compare to the spread of the data.

This means that the MSE presented in the results is actually not of how much actual "identity score" the model is off by, but rather of how many standard deviations of the training data it is off by. Since the training data has a rather low variance to begin with (in the order of thousandths), presenting only the "real" MSE would risk exaggerating the performance of the models. This also allows the experiment to be easily repeated and compared with using a different dataset. For clarity, the variance-dependent MSE is hereby referred to as *normal MSE*, and that MSE multiplied by the variance of the identity scores is simply referred to as MSE.

$$\text{Normal MSE} = \frac{\text{MSE}}{\sigma_s^2} \quad (5)$$

In order to verify that the produced estimator models are actually useful in predicting the identity score, their MSE was compared to the MSE of the constant estimator that estimates the identity score to be the mean of the whole dataset. If the MSE of the models are higher than the MSE of the constant estimator, that would indicate that the models are not useful since they would be less accurate at predicting the identity score than the constant estimator.

III. RESULTS

TABLE I
MSES OF THE BEST PERFORMING MODELS OF EACH TYPE

Estimator type	MSE	Normal MSE
Constant	$2.11 \cdot 10^{-3}$	1
Linear regression	$1.38 \cdot 10^{-3}$	0.65
General neural network	$6.12 \cdot 10^{-4}$	0.29
Conv. neural network	$2.18 \cdot 10^{-3}$	1.03

From the figures in table I, the performance of the different estimator types can be compared. The constant estimator, which would constantly estimate the identity score to be the mean value, had an MSE of $2.11 \cdot 10^{-3}$. For the non-normalized combined datasets, the mean of the identity scores was $\mu = 0.927$ and variance $\sigma^2 = 2.11 \cdot 10^{-3}$. As the MSE of unbiased constant estimator is always equal to the variance of the dataset, the normal MSE of it will naturally be 1. This will be the MSE value which the other models will be compared to.

The lowest normal MSE of 0.29 was achieved with a general neural network that used mean durations and mean alignment probability as input and 42 and 38 neurons for the first and second hidden layer, respectively. Several different neural network models were created and evaluated. Fig. 7 shows the learning curve of the best performing model. The fact that both training and validation loss is similar, indicates that the model is adept at predicting both test and training data. The results for the other neural networks using the same input features can be found in the appendix.

The best performing convolutional neural network achieved a normal MSE of 1.03, and consisted of 64 neurons each in both hidden layers. This signifies that the CNN model is worse at predicting identity score than even a constant estimator.

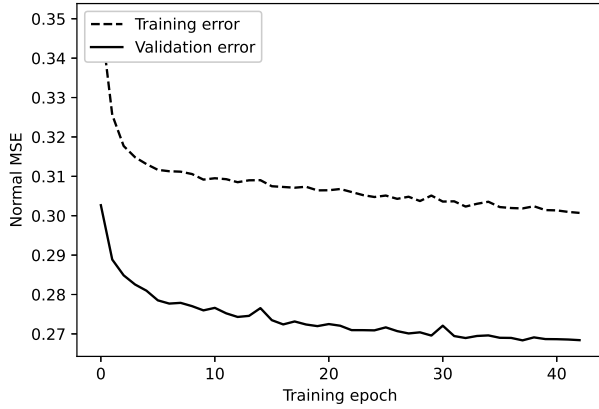


Fig. 7. Learning curve for the neural network that achieved the smallest MSE.

While this was not explicitly evaluated, using a higher learning rate for the SGD optimizer tended to result in a more irregular learning curve. For reference, compare the curve in Fig. 7, with a model using a learning rate of $1.2 \cdot 10^{-4}$, with the one found in Fig. 8, where the model used a learning rate of $1.2 \cdot 10^{-3}$.

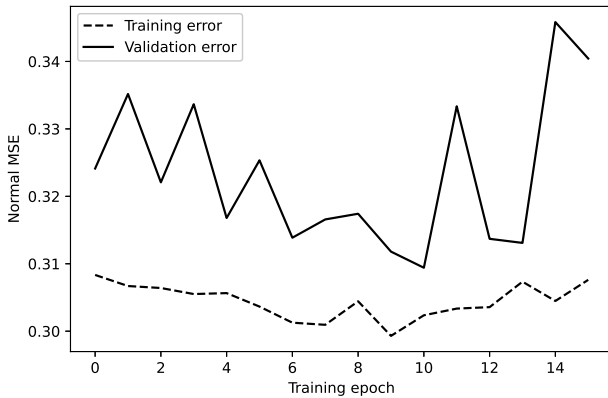


Fig. 8. An increased learning rate results in a more irregular learning curve.

The linear regression model that was created, shown in Fig. 9, had a normal MSE of 0.65. In Fig. 9 the line represents the linear approximation of the data and it is plotted on top of the 2D histogram shown in Fig. 4. In Fig. 9 it is possible to see that the linear model shows a positive correlation with the mean alignment probability. That is to say that the model predicts the identity score to increase for higher mean alignment probabilities. The correlation between the identity score and the mean alignment probability can also be seen in Fig. 3, the correlation matrix. The linear model is plotted on top of the 2D histogram in Fig. 4 to show the actual distribution of the samples which was used to create the linear regression model. It is possible to see from the 2D histogram alone that there is a positive correlation between the identity score and mean alignment probability.

The correlation matrix in Fig. 3, shows that the features that

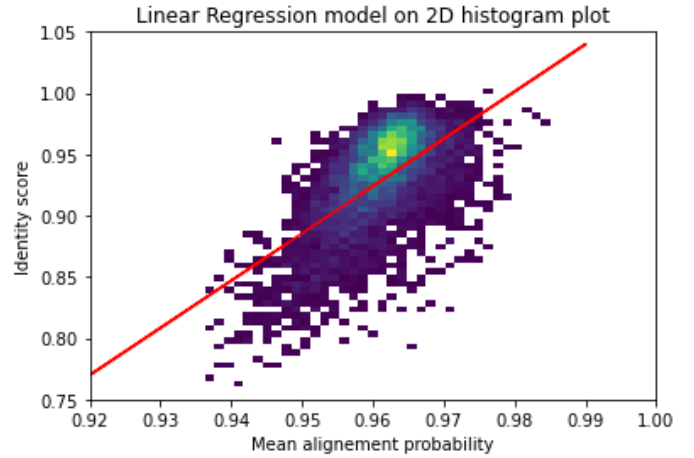


Fig. 9. Linear regression model plotted on top of a 2D histogram. The line shows the linear model between identity score and mean alignment probability. $MSE = 1.38 \cdot 10^{-3}$ and normal MSE = 0.65.

are the most correlated with the identity score are the mean alignment probability and standard deviation of alignment probability. After the alignment probabilities it is the mean duration, standard deviation of duration and the amount of samples of nucleobase T which were slightly correlated with the identity score. These features were tried as inputs to neural network models. It was discovered that not all of the features which were slightly correlated actually improved the neural network model.

In Fig. 10 it is shown the importance for each feature while using them as feature inputs for the neural network. The standard deviation of alignment probability was the most important feature, which could be expected according to the very high correlation in Fig. 3. However, the mean alignment probability which had an equally high correlation with identity score did not have a high feature importance. The mean and standard deviation of duration had a slight importance according to the feature importance plot, similar to what is shown in the correlation matrix in Fig. 3, however Fig. 10 shows that none of the other features had much of an impact at all. The correlation matrix indicates that the amount of samples of the base T has a similar correlation to the identity score as both the mean and standard deviation of duration, but the feature importance plot did not find it useful.

IV. DISCUSSION

The results show that the identity score of a sequence can be accurately and reliably predicted using no more than two input features. This makes it possible to train a satisfactory model using relatively few data points (as seen here) without the risk of overfitting.

The most important feature which was used by the neural network models was the mean alignment probability. That was also expected due to it having a very high correlation with the identity score, seen from the correlation matrix in Fig. 3. Interestingly enough, there were discrepancies between the relative feature importance extracted from the neural network training in Fig. 10, and the correlation matrix in Fig. 3.

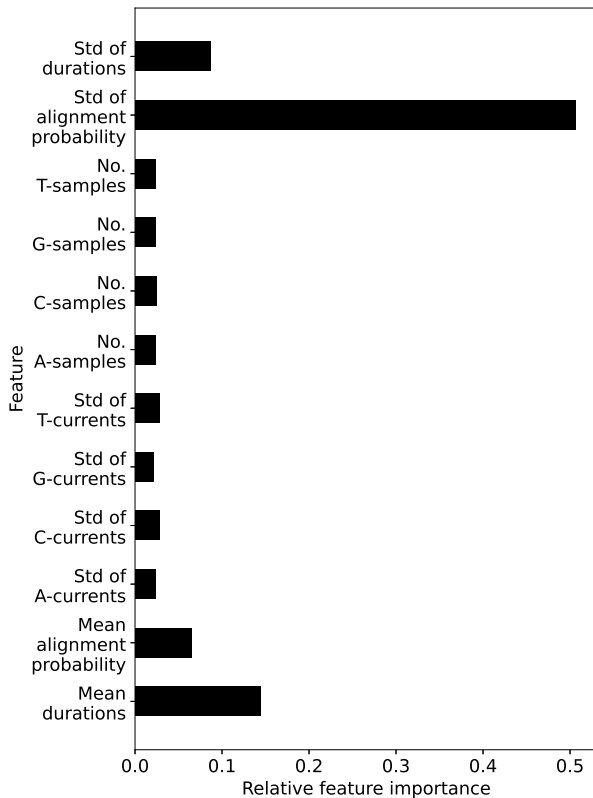


Fig. 10. The relative importance of the features when used as training data.

The plot for the relative feature importance shows that the most important feature is the standard deviation of alignment probability. The correlation matrix, however, shows that both the mean alignment probability and standard deviation of alignment probability were equally correlated to the identity score. In fact they were almost completely correlated to one another with a correlation of -0.98 . Furthermore, the mean alignment probability and mean duration were the two features which created the best neural network model.

The high correlation between the two alignment probabilities could be the reason for why the plot of the relative feature importance, shown in Fig. 10, attributes a smaller level of importance to the mean alignment probability. Due to their very high correlation, the two alignment probabilities provides similar information in regards to predicting the identity score.

For example, if the standard deviation of alignment probability is already used as a feature to predict the identity score, adding the mean alignment probability to that model would not bring any new information in predicting the identity score. Once the decision tree has already decided to split using the standard deviation of alignment probability to improve the model, it will then ignore splitting with regards to the mean alignment probability since that does not bring any new information to the model. This is most likely the reason for the discrepancy in alignment probabilities between the correlation matrix in Fig. 3 and the feature importance plot in Fig. 10.

To further investigate the differences between the correlation matrix and relative feature importance, another relative feature

importance plot was created. When a similar feature importance plot to Fig. 10 was created without the standard deviation of alignment probability, the mean alignment probability then became the most important feature. It had assumed the same relative feature importance that the standard deviation had in Fig. 10. This confirms that the mean alignment probability is as important as the standard deviation of alignment probability in predicting the identity score, but that the reduced importance seen in Fig. 10 is due to their high correlation with one another. The result is consistent with what was shown in Fig. 3.

Because of how important the alignment probability is in predicting identity score, it would be interesting to investigate other properties relating to it in future work.

It is unclear why the convolutional neural networks performed so poorly. One likely explanation is that it simply was not properly implemented into the software. Given enough time to debug and/or tweak parameters in the models, a better result may be obtained using CNNs.

One concern which could be investigated further in future research projects is whether these models are able to predict the identity scores of DNA samples from other organisms. The data that these models were trained on were DNA from *E. coli* bacteria, so there is the possibility that these models are overfitted to *E. coli* DNA. In a similar vein, there is also the concern for overfitting in regards to the base calling algorithm used to produce the identity scores. Due to the fact that these models were trained to predict identity scores from one specific base calling algorithm, there is a chance that the models shown in this report might not be a reliable predictor for other base calling algorithms. Additionally, if another base calling algorithm does not output the same features as the one used in this project, most of the models described in this report would not work at all.

There are a remarkable number of parameters that may have an impact on the performance of the model, that were not systematically evaluated for this project. Among these are the learning rate of the SGD optimizer, the activation functions used for the hidden layer neurons, and the conditions for triggering an early stop.

To evaluate how well the models performed, the only metric used in this project was the mean square error of the trained model. That is to say that the only thing that indicated whether one model was a better choice than the other, was simply their respective error rates. While this does lead to a better performing final model overall, this approach made training computationally expensive. This was because the models that provided the best results were often the ones that required more time to train (i.e. neural networks with lower learning rates). In the future, making a point of minimizing, or at least monitoring the complexity of training may lead to faster improvements.

V. CONCLUSION

In this project the goal was to create a model which could accurately predict the identity score of a DNA sample sequenced through nanopore sequencing. The identity score is the accuracy rate of the sampled DNA compared to a reference

sequence of the DNA. The best model created was a general neural network that used mean durations and mean alignment probability as input and 42 and 38 neurons for the first and second hidden layer, respectively, with an MSE of $6.12 \cdot 10^{-4}$, compared to a variance of $2.11 \cdot 10^{-3}$.

Neural networks with nanopore data as inputs were shown to be able to reliably predict the accuracy rate of the sequenced data. This will be useful in aiding DNA base calling used in nanopore sequencing, by being able to align multiple sequences with one another using identity score as an indicator.

APPENDIX

NORMAL MSEs FOR GENERAL NEURAL NETWORKS

ACKNOWLEDGMENT

The authors would firstly like to thank the project supervisors for their consistent patience and guidance throughout the project. Secondly, they would like to thank both of their respective families for their constant support during the last term. Finally, Milad would like to extend additional thanks to his good friends Emil, Leon and Hana, for giving the report one final proofreading.

REFERENCES

- [1] T. Hu, N. Chitnis, D. Monos, and A. Dinh, "Next-generation sequencing technologies: An overview," *Human Immunology*, vol. 82, no. 11, pp. 801–811, 2021, next Generation Sequencing and its Application to Medical Laboratory Immunology. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0198885921000628>
- [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, 07 2018.
- [3] E. I. N. News and S. Foxton, "Oxford Nanopore announces single-read accuracy of 99.1% & sequencing a record 10 Tb of DNA in a single PromethION run," *EIN News*, Dec. 2020. [Online]. Available: https://www.einnews.com/pr_news/531976603/oxford-nanopore-announces-single-read-accuracy-of-99-1-sequencing-a-record-10-tb-of-dna-in-a-single-promethion-run
- [4] N. Huang, F. Nie, P. Ni, F. Luo, and J. Wang, "Sacall: A neural network basecaller for oxford nanopore sequencing data based on self-attention mechanism," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 19, no. 1, pp. 614–623, 2022.
- [5] M. Jain, I. T. Fiddes, K. H. Miga, H. E. Olsen, B. Paten, and M. Akeson, "Improved data analysis for the minion nanopore sequencer," *Nature methods*, vol. 12, no. 4, pp. 351–356, 2015.
- [6] J. Kipen, private communication, May 2022.
- [7] Oxford Nanopore Technologies. (2022, Jan.) Scientists describe new approach in nejm, using oxford nanopore dna sequencing technology to improve prognosis in critically ill patients, in less than 8 hours. [Online]. Available: <https://nanoporetech.com/about-us/news/scientists-describe-new-approach-nejm-using-oxford-nanopore-dna-sequencing-technology>
- [8] D. Joshi, S. Mao, S. Kannan, and S. Diggavi, "Qalign: Aligning nanopore reads accurately using current-level modeling," *bioRxiv*, 2019. [Online]. Available: <https://www.biorxiv.org/content/early/2019/12/03/862813>
- [9] D. O. Wackerly, W. Mendenhall, and R. L. Scheaffer, *Mathematical statistics with applications, international edition*, 7th ed. Florence, KY: Brooks/Cole, Oct. 2007.
- [10] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," 2016, arXiv identifier: arXiv:1607.06450v1. [Online]. Available: <https://arxiv.org/abs/1607.06450>
- [11] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6.

CONTEXT O

ARTIFICIAL INTELLIGENCE

POPULAR DESCRIPTION

No quiet on the AI frontier - How chess, AI and war are connected

Doomsday is near, the day when AI will conquer humans - that was what many of us thought two decades ago. The growth of AI in multiple areas including games has fascinated and fascinates people around the world on a daily basis. Soon, the capabilities of humans will be tiny compared to those of AI. In fact, in some cases the computers are already outperforming us humans.

When Garry Kasparov was beaten in chess by the AI Deep Blue in 1997, many people saw this as the ultimate accomplishment of computers in games. Since then, the development of artificial intelligence in general, and its applications in games particularly, has continued at an astonishing speed. Games that were earlier considered unsolvable, and where man was thought to have the upper hand, have in the last couple of years also become dominated by AI-algorithms. An example of this is OpenAI's artificial Go-player who beat the then world champion, Lee Sedol, in 2016.

While AI has managed to beat us in strategic games such as chess across the board, the possibilities in real world problems can still be explored further. A game is often phenomenologically coupled to what we normally perceive as games, such as the ones mentioned above. However, games can also be representations of conflict, and what conflict is bigger than war? Thus, wars can be called the biggest of games and the most complex game ever played and every game is a small-scale war and every game that is solved brings us one step closer to the perfect war strategy.

There is no reason to think this development will slow down, so in the coming years expect AI to continue getting smarter. While it is all fun and games right now, we need to proceed with caution or humanity might be the one getting outplayed.

SUMMARY OF THE PROJECT RESULTS

In the last few years, the performance of Artificial Intelligence (AI) within games has improved drastically and AI-bots have been winning over human players in more and more games. One interesting aspect of creating and researching AI for games is to observe and analyze its ability to develop effective strategies to play games. This acquired knowledge can afterwards be used to solve real problems with similar constraints as the games and even ease development of solutions where constraints are different.

One topic, which was studied in projects O1a and O1b, is multiplayer games. Multi-agent games of imperfect information are a kind of games where a coalition of agents aim to fulfill some set of objectives against some sort of opponent, where imperfect information means that agents might not be able to distinguish certain game states from others. There are several key concepts linked to this group of games. Among these are finding strategies and the concepts of knowledge. Finding a strategy for the coalition of agents to achieve a common objective is an area with a plethora of research possibilities. Furthermore, when finding these strategies, the knowledge of the individual agents is instrumental in devising effective strategies.

In project O1a, grid based multiplayer pursuit evasion games with rational agents were studied where pursuers can share their knowledge during certain circumstances. To begin with, we formalized pursuit evasion games and the concept of knowledge within these games. This formalization aimed primarily to explain the concept of order of knowledge. Essentially, knowledge of higher order describes how agents may use deduction to draw conclusions about the current state of the game.

Finally, project group O1a developed and evaluated a higher-order knowledge-based strategy within the formalized pursuit evasion game and concluded that there is little to no difference in the effectiveness of different order strategies.

In project O1b games with imperfect information were studied, where a team of agents tries to reach a common objective when playing games versus nature (or a second player). The agents were divided into two subgroups, agents with and without strategies. All these strategies were considered common knowledge. The project goal of O1b was to formulate a translation of the original game to a modified game where agents with the predefined strategies (PDS) were considered redundant information because of their predictability. This “projection tool” would permit the project group to synthesize strategies for all the agents without a predefined strategy using a tool called “Multi-Agent Knowledge-Based Subset construction” (MKBSC). This tool is based on a concept that higher levels of knowledge can be gathered when iterating over the game, to generate increasingly higher levels of knowledge, which was used to find a knowledge state in which a winning strategy is found.

The groups in Project O2 have implemented a game playing AI using Monte Carlo Tree Search (MCTS) for the 2-player board game - Fox Game. MCTS is an algorithm that finds the most promising move by simulating many games where random moves are made until either player wins. Afterwards, the results are back propagated. Both groups implemented a cut-off, which is an optimization technique where the random games are ended prematurely, and the board state is instead evaluated using an evaluation function.

Fox Game is a board game that has been around in different forms since the 15th century. The game is a two-player game where one player plays as 20 sheep with the goal to traverse the game board into a pasture, and the other player plays as two foxes that are trying to prevent the sheep from reaching their goal. As the game has been around for a very long time in many different forms there are many variations of the rules, where none can be considered neither correct nor false. As it is not a very well-known game, there is no prior online adaptation of Fox Game and no AI-implementation either.

In project O2a, the performance of the MCTS algorithm was compared to another algorithm called alpha-beta search, which looks a few moves ahead and selects the move which leads to the best game state after those moves given that the opponent plays the best responses. They were tested against each other with several different turn timers to see if the time has a significant effect on their performances.

The group in O2b focused on optimizing the evaluation function, and thereby finding an optimal strategy for the Fox Game. The structure of the evaluation function was inspired by more established board games such as chess, and contains a material part, which considers the relative value of the different pieces, and a piece-square table, which assigns a value to each board space. The optimal values of the constants in the evaluation function were found by having agents with different values compete against each other until a set of superior constants was found. The results discovered by the group gives insights into how the foxes should be valued in relation to the sheep, as well as how the material part and the piece-square table should be valued in relation to each other.

The results from O2a give insights in which games an implementation of MCTS could be a promising approach. The results from O2b give insights in how to evaluate different board states in the Fox Game and thereby what actions are favorable.

One area that would be interesting to continue exploring in project O1a is to examine if it is possible to formulate knowledge representations of higher order within grid-based pursuit evasion games where pursuers are unable to communicate. It would also be of interest to investigate if it is possible to apply the multiplayer knowledge-based subset construction to simple examples of pursuit evasion games. Possible further work on the subjects that project O1b explores could include research of conditions that allow certain games to stabilize with higher order of knowledge and to see if this can be implemented in code. Another research area is to more formally and rigorously define the concepts introduced in the projects. Another line of future work that would be interesting to pursue is to use neural networks in combination with MCTS to make a more precise evaluation function for Fox Game, and in extension a more competent AI.

IMPACT ON SOCIETY AND ENVIRONMENT

The rise of AI in the production and transport sectors offers a lot of opportunities for optimization regarding both energy and material management. Small gains can be made in everything from self-driving trucks to real time management of energy production by using AI. Even though every single one of these improvements might be insignificant, the total sum can have a huge impact on the environment by reducing energy consumption during all stages of the transportation process. AI could also be used for research purposes to find currently unknown technologies or implement brand new ones which would benefit the environment greatly.

Today we are starting to see an increase in the implementation of AI decision-based products in everyday life, like, for example in our cars, homes, phones and other apparatus, making our homes smart, our phones more intelligent and our cars so autonomous that they are almost able to drive themselves. These improvements have started a revolution, but the changes are not only contained to personal improvements but are seen throughout different business sectors. AI has the potential to automate production and other manual labor sectors as well. Depending on your view of society and your personal life, this change may either be regarded as good or bad. One obvious negative effect here is that it would lead to more jobs being taken by robots, either entirely or partly. Depending on how society reacts to this, both politically and culturally, this may either lead to mass unemployment and unhappiness, or shorter workdays that could increase the well-being and health of workers. This could also free up people's time so that more people can work within other fields where there currently is a shortage of workers, such as health care. The transition of the workforce that would occur if all Amazon warehouse employees were replaced by autonomous robots, could lead to a large societal impact. A big implication would be that when all humans are replaced, there could be a surge in effectiveness of order-to-delivery time. These robots could work independently of communication between each other and internet connections, making these robots more reliable and robust. The new implemented network of AI robots would work with continuous expansion, where implementing newer, more advanced robots would not require the retirement of old robots. This makes the transition more reliable and cost effective, allowing the transportation, merchandise sector to see larger savings and expansions. This would however require society to adapt and find new ways to employ and stimulate workers and inhabitants. We believe that this adaptation would not necessarily be an easy one, where larger rates of unemployment could lead to greater decrease in prosperity and remove the feeling of belonging that is associated with working. These unemployed individuals would also face considerable financial problems unless governments introduce welfare systems with the purpose of helping these individuals.

The development of strategies under different conditions of imperfect information and within various contexts is treated by both projects O1a and O1b and could have multiple applications in society and propel the usage of AI. One area where this could play a significant role is in the context of search and rescue missions. A coalition of firefighting drones could coordinate its actions with the help of knowledge-based strategies based on information perhaps gained and interpreted from area footage or temperature and weather measurements in order to distinguish the fire as quickly as possible. AI could also be used to perform rescue missions when searching for lost people and coordinate such missions where actors and areas are known but uncertainty about other things such as positions are present. For a benevolent society like above, saving people and managing forest fires more effectively are unarguably good things for both individual people as well as the environment.

However, governments with ill intent may misuse AI with the purpose of controlling the people. For instance, they may coordinate intelligent agents in order to locate political opponents and unwanted individuals using for example cameras and social-media surveillance. We have already seen traces of this in Chinese society where the government is using knowledge-based strategies to optimize their search for people they see as threats to the nation.

Context O is about AI applied to game playing. At first glance, the topic might seem harmless and fun but future possible applications reveal that there are consequences that should not be taken lightly. One example originates from the collaboration of project group O2 and Swedish Defense Research Agency (FOI) where a game-playing AI is developed. FOI has shown interest in applying the knowledge gained from this research in different types of strategic decisions, most prominently in warfare. If such knowledge were applied, it could help strategic decisions in military aggression. In the case of defensive actions, more lives can be saved. Future applications of this research are unknown, but it is still important to discuss their possible positive and negative impacts on society and individuals.

Grid-based Pursuit Evasion Games of Imperfect Information: Theory and Higher Order Knowledge-based Strategies

Jacob Granqvist and Jonas Haker

Abstract—One group of games studied within game theory are grid-based pursuit evasion games of imperfect information. A pursuit evasion game is in essence a game where there exists a set of pursuers which have as their objective to capture a set of evaders. This thesis aims to develop a formalisation of this type of games as well as describing and integrating vital game theoretical concepts such as order of knowledge into this game. With the developed formalism at hand, the concept of knowledge-based strategies is then introduced, which is essential when searching for the way to play the game most efficiently. The formalisation of the game is then followed by a simulation, measuring the performance of some older and some newly developed knowledge-based strategies. The thesis concludes that the formalisation is applicable on a more general class of pursuit evasion games and enables a wider study of the game. The simulation results indicate that knowledge-based strategies of higher order do not always perform better compared to simpler strategies of lower order of knowledge. Furthermore, strategies which allow for communication between agents are found to be superior to communication-less strategies.

Sammanfattning—En typ av spel som studeras inom spelteori är rutnätbaserade jakt-flykt-spel med ofullständig information. Ett jakt-flykt-spel går ut på att det existerar en samling jagande aktörer som försöker fånga en samling flyende aktörer. Denna uppsats söker utveckla en formalism för denna typ av spel såväl som att beskriva och integrera ett antal nyckelkoncept inom spelteori såsom kunskapsordning. Med hjälp av den utvecklade formalismen, framställs så kallade kunskapsbaserade strategier, vilka är av fundamental vikt i sökandet efter sätt att spela spelet på det effektivaste sättet. Kapitlet om formalismen följs sedan av simuleringar där några äldre och några nyare kunskapsbaserade strategier prövas. Slutsatsen dras att den nya formalismen kan vara applicerbar på en bredare samling jakt-flykt-spel än den initialt påtänkta. Vidare underlättar formalismen en generalisering till andra sätt att beskriva spel. Simulationsresultaten indikerar att kunskapsbaserade strategier av högre ordning inte alltid presterar bättre än enklare strategier av lägre ordning. Till yttermera visso visar sig kommunikationslösa strategier vara underlägsna strategier som tillåter kommunikation.

Index Terms—Pursuit Evasion Games, Knowledge representation, Imperfect Information, Higher Order Knowledge, Knowledge-based Strategies, Communication-based Strategies, Game Theory.

Supervisors: Dilian Gurov

TRITA number: TRITA-EECS-EX-2022:179

I. INTRODUCTION

Game theory describes how mathematical models can be used to study interactions between rational agents and is

widely used within for example economics, computer science and philosophy. Many kinds of strategic interactions in society can be viewed as games, be it movement of armies in wars or automated robots in a factory. Consequently, there exists a plethora of games which can be studied within game theory that can have real world applications. One class of games that has been widely studied are pursuit evasion games (henceforth called PEG:s). The basic premise of this type of game is that we have a set of pursuers tasked with chasing and capturing a set of evaders. The type of PEG we shall study in this thesis is a discrete turn based PEG played on a finite grid.

The aim of this thesis is to formally define PEG:s of the above kind. We will also argue how this representation combined with a concept known as higher-order knowledge may be used to extract knowledge-based strategies. Finally, we will present a strategy based on higher-order knowledge and with the help of simulations determine its effectiveness in comparison with other strategies of lower order.

II. PURSUIT EVASION GAMES

A PEG is a type of game where a number of pursuers have as their objective to find and hunt down one or more evaders [1]. Evidently, this definition gives room for a lot of variability when it comes to setting up the rules for a PEG. Below we will detail the PEG which we intend to study in this thesis. First we will formally define a PEG and exemplify its properties. Then, we will continue by describing how pursuers may use deduction to extract information about the current state of the game. We will continue with explaining how the knowledge about the current affairs may be represented using a knowledge representation. By investigating this representation, we may then construct knowledge-based strategies for the pursuers.

A. Formal definition of PEG on finite grid

Let us formally define the game which is of interest in this thesis. A PEG is played on a map

$$\mathcal{M} = \langle V_n, E_m, E_v, E_c \rangle$$

where $V_n = I_n \times I_n$ is a set of sites. I_n is a finite set of points defining the axes

$$I_n = \{0, 1, 2, \dots, n-1\}.$$

E_m, E_v and E_c will be described in the upcoming subsections. A location $loc \in V_n$ is a site (x, y) where x denotes the column and y denotes the row.

The game is played by sets of pursuers

$$A_p = \{p_i\}_{i=1}^l, \text{ some } l \in \mathbb{N}$$

and evaders

$$A_e = \{e_i\}_{i=1}^k, \text{ some } k \in \mathbb{N}$$

together forming the set of agents

$$\mathcal{A} = A_p \cup A_e.$$

Observe that the set A_e is a dynamic set where an evader $e \in A_e$ is removed from the set whenever it is captured. Continuing, the set

$$E_m \subseteq V_n \times V_n$$

defines a move relation between two locations in V_n . The movement relation E_m restricts the possible movements an agent can perform. On finite grids, a movement can be performed only to one of the directly neighbouring locations in V_n . It is defined as below:

$$\text{If } (x, y) \text{ and } (x', y') \in V \text{ then } ((x, y), (x', y')) \in E_m \iff (x' = x \pm 1 \wedge y' = y) \vee (x' = x \wedge y' = y \pm 1)$$

As can be understood this relation is a symmetric relation, which means that movement between positions in V_n go both ways. The definition does, intentionally, not allow for diagonal movement.

Throughout this thesis, it will be assumed that evaders always move randomly. Consequently, they will not necessarily choose a legal move that will benefit them in avoiding the pursuers.

In the definition of the map above, E_m and V_n together define the grid

$$G_n = \langle V_n, E_m \rangle$$

upon which the PEG is played.

With this definition, a PEG can be played on a $n \times n$ -matrix where moves between locations in V_n are simply transitions to an adjacent location in a matrix. The matrix representation will onward be used to exemplify the different properties of the PEG.

Example 2.1: Following is an example of a legal move for an agent $a \in \mathcal{A}$ as prescribed by E_m on a 4×4 matrix. As for now, we denote locations in V_n that are not occupied by an agent with a 0. Henceforth, arrows will be used to clarify and denote the direction of the movement performed by an agent.

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & a & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \longrightarrow \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \rightarrow & a \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

△

Two other relations which will be of great importance are the visibility relation E_v and the communication relation E_c . The set

$$E_v \subseteq V_n \times V_n$$

defines a visibility relation which will be defined at a later stage in the chapter on imperfect information. Likewise, we refrain from defining the communication relation

$$E_c \subseteq A_p \times A_p$$

for now and leave it for the chapter on rules of communication.

A state s of a game on the map \mathcal{M} is an ordered pair

$$s = (pos, turn)$$

where

$$pos : \mathcal{A} \rightarrow V_n$$

maps every active agent of the game to a location on the grid. The set

$$turn \in \{p = \text{pursuers' turn to make moves}, \\ e = \text{evaders' turn to make moves}\}$$

specifies which set of agents should make the next move. We define the set S to be the set of all possible game states.

Example 2.2: A sample of two different game states, s and s' , using the matrix representation.

$$s = \left(\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & p_1 & 0 & p_2 \\ 0 & 0 & e_2 & 0 \\ 0 & e_1 & 0 & 0 \end{bmatrix}, p \right)$$

$$s' = \left(\begin{bmatrix} 0 & p_1 & 0 & p_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & e_2 & 0 \\ 0 & e_1 & 0 & 0 \end{bmatrix}, e \right)$$

△

With the above given, we can now finally define a PEG as the tuple

$$\mathcal{G} = \langle \mathcal{M}, A_p, A_e, s_0 \rangle$$

where s_0 is the initial state, a configuration of the *initial* agents on the grid.

Furthermore, we can define a history \mathcal{H} at time point

$$t \in \mathbb{N}$$

to be a finite sequence of states

$$\mathcal{H} = (s_0, s_1, s_2, \dots, s_t)$$

collecting every move in a game up to and including state t .

B. Rules of the game

The rules of the Pursuit Evasion game we intend to study are essentially the same as the rules used by Goobar and Söderberg in [2].

- 1) The PEG is carried out through alternating moves. The pursuers begin making moves. All agents of a certain type move simultaneously. Additionally, to remain in the same position, i.e. not to move, is not a legal move.

Example 2.3: A PEG-played on a 4×4 grid with $|A_p| = 2$, $|A_e| = 2$ and the initial state

$$s_0 = \left(\begin{bmatrix} 0 & 0 & 0 & p_2 \\ p_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & e_2 \\ 0 & e_1 & 0 & 0 \end{bmatrix}, p \right).$$

The pursuers start by making moves, and the game transitions to a new state

$$s_1 = \left(\begin{bmatrix} 0 & 0 & p_2 & \leftarrow \\ \downarrow & 0 & 0 & 0 \\ p_1 & 0 & 0 & e_2 \\ 0 & e_1 & 0 & 0 \end{bmatrix}, e \right)$$

which is followed by the moves of the evaders into the state

$$s_2 = \left(\begin{bmatrix} 0 & 0 & p_2 & 0 \\ 0 & 0 & 0 & 0 \\ p_1 & e_1 & 0 & \downarrow \\ 0 & \uparrow & 0 & e_2 \end{bmatrix}, p \right).$$

△

- 2) Multiple agents of the same type can occupy the same locations
- 3) An evader is caught and removed from the game if at the end of a turn it occupies the same position as a pursuer.

Example 2.4: Observe the following state s_k of a PEG played upon a 4×4 grid with $A_p = \{p\}$ and $A_e = \{e_1, e_2\}$:

$$s_k = \left(\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & p & 0 \\ e_2 & 0 & e_1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, p \right).$$

The pursuer proceeds by moving into the same position as the evader and eliminates the evader such that

$$s_{k+1} = \left(\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \downarrow & 0 \\ e_2 & 0 & p & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, e \right).$$

△

By this definition, evaders can also be eliminated by moving into the same position as a pursuer.

- 4) The objective of the game is for the pursuers to capture all evaders by occupying every evader's grid site (and thereby capturing the evader in that site) at some point of the game
- 5) A game is said to be won when every evader $e \in A_e$ has been eliminated. Namely,

$$A_e = \{\emptyset\}.$$

C. Imperfect information

When playing a game, there might be information about the current state of the game that is either partially or completely hidden for the agents. This is known as a game of imperfect information. Firstly, we need to define what type of information should be considered commonly known in the PEG and what type of information should be partially or completely unknown for the pursuers.

Some information about the game and game structure are defined to be *common knowledge*.

1) *Assumed as common knowledge:*

- The number of agents in play
- Rules of the game
- The size of the grid.

Other information is defined to be *unknown* or *partially unknown*.

2) *Assumed to be unknown or partially unknown:*

- The positions of other agents.

The knowledge about positions of other agents are essential in our thesis, and brings us to explain the meaning of the set $E_v \in \mathcal{M}$ which we abstained from defining earlier. Firstly, we need to explain the concept of visibility. A location loc is said to be visible to a pursuer p if

$$(pos_p, loc) \in E_v.$$

As for now, we denote a visible location in the state matrix with a 0, and positions that are not visible with a *. The visibility relation E_v can now be constructed in several different ways. We shall in this thesis focus on one type of visibility, namely corridor-based visibility. However, two different types of visibility relations will be exemplified below.

- 1) *Corridor-based visibility:* Every location on the same row or column as the pursuer is visible to the pursuer.

$$\text{If } (x, y) \text{ and } (x', y') \in V \text{ then } ((x, y), (x', y')) \in E_v \iff (x' = x) \vee (y' = y)$$

Example 2.5: Visible locations for a pursuer based on the corridor-based visibility definition.

$$\begin{bmatrix} * & 0 & * & * \\ * & 0 & * & * \\ 0 & p & 0 & 0 \\ * & 0 & * & * \end{bmatrix}$$

△

- 2) *Radius-based visibility:*

We must first define the distance on our map. In this context, it is natural to define the distance as the metric in \mathbb{R}^2 . $d(p, q) = \sqrt{|p_x - q_x|^2 + |p_y - q_y|^2} = d$.

$$\begin{aligned} &\text{If } (x, y) = \bar{x} \text{ and } (x', y') = \bar{x}' \in V \\ &\text{then } ((x, y), (x', y')) \in E_v \iff \\ &d(\bar{x}, \bar{x}') \leq r \end{aligned}$$

for some r .

Example 2.6: Visibility for a pursuer based on the radius-based visibility definition with $r = 1$.

$$\begin{bmatrix} * & * & 0 & * \\ * & 0 & p & 0 \\ * & * & 0 & * \\ * & * & * & * \end{bmatrix}$$

△

By restricting the visibility of the pursuer, the concept of indistinguishable states naturally arises. Two states are said to be indistinguishable for a pursuer p if the pursuer cannot tell them apart.

Example 2.7: Assume corridor-based visibility. Given that a certain sector of the grid is unobservable, as defined by the corridor visibility relation, we obtain imperfectness concerning the knowledge about the other agents whereabouts. Thus, multiple states of the games become indistinguishable for p_1 . Observe the following example of indistinguishable states for pursuer p_1 :

$$\begin{pmatrix} \begin{bmatrix} 0 & 0 & e & 0 \\ 0 & p_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ p_1 & 0 & 0 & 0 \end{bmatrix}, p \end{pmatrix} \sim_1 \begin{pmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & p_2 & 0 & 0 \\ 0 & 0 & e & 0 \\ p_1 & 0 & 0 & 0 \end{bmatrix}, p \end{pmatrix} \sim_1 \begin{pmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & e & p_2 & 0 \\ 0 & 0 & 0 & 0 \\ p_1 & 0 & 0 & 0 \end{bmatrix}, p \end{pmatrix} \sim_1 \begin{pmatrix} \begin{bmatrix} 0 & p_2 & 0 & 0 \\ 0 & 0 & e & 0 \\ 0 & 0 & 0 & 0 \\ p_1 & 0 & 0 & 0 \end{bmatrix}, p \end{pmatrix} \text{ etc.}$$

△

In the example above, the set of indistinguishable states consists of all possible legal configurations of the other agents' positions in the top right of the grid (star marked area below).

$$\begin{bmatrix} 0 & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \\ p_1 & 0 & 0 & 0 \end{bmatrix}$$

Since pursuers and evaders are not allowed to occupy the same locations, the number of different indistinguishable states amount to

$$9 \times 8 = 72$$

different states.

D. Rules of communication

If we assume the pursuers are rational agents it is important to clarify if and when pursuers are allowed to communicate. The rule of communication clarifies when pursuers can communicate, and is related to a concept known as knowledge which will be explained later. Following are two ways to define when knowledge will be shared:

- 1) Knowledge is never shared directly to other pursuers.
- 2) The pursuers knowledge is shared with other pursuers whenever they are visible to one another.

The concept of communication can be generalised with the equivalence relation

$$E_c \subseteq A_p \times A_p$$

which defines a communication relation between two pursuers in A_p . The communication relation E_c determines which pursuers can share information with one another. We define the equivalence class $[p]$ to be the set of all pursuers that can communicate with p . Namely,

$$[p] = \{p_i \in A_p : (p, p_i) \in E_c\}.$$

Notice that these are dynamic equivalence classes since pursuer communicate with different sets of pursuers after every move. If we follow the first rule of communication we define

$$E_c = \{(p, p) : p \in A_p\}$$

which implies that pursuers can only communicate with themselves. On the contrary, the relation will be of the following form if the second rule of communication is used:

$$\text{Take } p_1, p_2 \in A_p$$

$$\text{If } (pos_{p_1}, pos_{p_2}) \in E_v \implies (p_1, p_2) \in E_c$$

Note the implication. Two pursuers might not be able to observe one another directly but be able to communicate with each other indirectly through a chain of pursuers.

In this thesis, we shall investigate how these rules may be used when simulating a PEG.

E. Observations

Another important concept which we will need are observations. An observation $o_p \in \mathcal{O}_p$ of a pursuer is given by the visibility relation E_v and defined as all states $s \in S$ that pursuer p cannot distinguish from each other. The set \mathcal{O}_p is the set of all possible observations of a pursuer p and partitions S , the set of states of the game. This definition follows from [3]. That is, S is comprised of the disjoint union of the elements in \mathcal{O}_p .

Example 2.8: Observe the following state of a PEG following the corridor-based visibility definition with $A_p = \{p_1, p_2\}$ and $A_e = \{e\}$.

$$s = \left(\begin{bmatrix} 0 & p_2 & 0 \\ 0 & 0 & 0 \\ 0 & e & p_1 \end{bmatrix}, p \right)$$

Obviously, the four states

$$s = \left(\begin{bmatrix} 0 & p_2 & 0 \\ 0 & 0 & 0 \\ 0 & e & p_1 \end{bmatrix}, p \right), s' = \left(\begin{bmatrix} p_2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & e & p_1 \end{bmatrix}, p \right)$$

$$s'' = \left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & p_2 & 0 \\ 0 & e & p_1 \end{bmatrix}, p \right), s''' = \left(\begin{bmatrix} 0 & 0 & 0 \\ p_2 & 0 & 0 \\ 0 & e & p_1 \end{bmatrix}, p \right)$$

are indistinguishable for pursuer p_1 and the set

$$\{s, s', s'', s'''\}$$

constitute all indistinguishable states for p_1 . This set therefore form an observation

$$o_{p_1} = \{s, s', s'', s'''\} \in \mathcal{O}_{p_1}$$

for p_1 . Another way of writing the same observation is by using the representation

$$o_{p_1} = (\mathcal{I}, turn) = \left(\begin{bmatrix} * & * & 0 \\ * & * & 0 \\ 0 & e & p_1 \end{bmatrix}, p \right)$$

where the indistinguishability matrix \mathcal{I} describes all agent configurations that are indistinguishable for the pursuer.

△

The indistinguishability matrix is a way of simplifying the notation for the possible states of the unobserved agents. These states consist of the possible permutations of the unobserved agents on the unobserved locations. The set of their positions can be represented by a cartesian product between the possible

positions of these agents. However, some pruning needs to be done in order to account for the impossible states where a pursuer and an evader occupies the same position. That a cartesian be used comes from the fact that the possible configurations of the unknown positions are homogenous (the position of a pursuer does not affect the position of an evader and vice versa) and the states are therefore orthogonal.

F. Knowledge representation

Continuing, every pursuer has a set of knowledge about the possible states of the game. A knowledge representation is a structure containing the current knowledge of a pursuer p . The knowledge representation will be defined differently depending on the order of knowledge.

Definition 1 (Order of knowledge): The order of knowledge for an agent is given inductively by the definition in [4].

- Order-0 knowledge: Knowledge about what the agent currently senses. No deduction about the current state-of-affairs has been made. Represented using **sets** of observations.
- Order-1 knowledge: Most precise estimate of the current state of affairs for the agent. Represented using subsets of observations.
- Order-(k+1) knowledge: Includes order-1 knowledge about themselves and possible order k knowledge of the other players.

△

G. Knowledge Representation in Pursuit Evasion Games

Using the previously presented definition, the knowledge representation \mathcal{K} for a pursuer p of order $(k + 1)$, at time point t , is given using induction. An instance of a knowledge representation is called a knowledge state.

Base Cases:

$$\mathcal{K}_p^0(t) = \{\text{possible states of the game}\} \in \mathcal{O}_p$$

$$\mathcal{K}_p^1(t) = \{\text{deduced possible states of the game}\} \subseteq \mathcal{O}_p \in \mathcal{O}_p$$

Inductive step:

$$\mathcal{K}_p^{k+1}(t) = \{\mathcal{K}_p^1(t), \mathbb{K}_{\text{all other pursuers}}^k(t)\}$$

Where

$$k \in \mathbb{N}^+$$

and

$$\mathbb{K}_{\text{all other pursuers}}^k(t)$$

is defined to be all possible knowledge states of the other pursuers of order k .

Looking back on example 2.8, it can now be established that \mathcal{O}_{p_1} is the order-0 knowledge of p_1 . In fact, since there exists no history from which to deduct, this is also the order-1 knowledge of p_1 . $\mathcal{K}_p^0 = \mathcal{K}_p^1 = \mathcal{O}_{p_1}$.

H. Knowledge sharing

If two or more pursuers can communicate with one another as defined by the second communication relation their knowledge will be intersected and the amount of possible states will be reduced. The shared knowledge set, which is the knowledge available to all pursuers $p_i \in [p]$, is generated in the following manner. Take $p \in A_p$. We then have

$$\mathcal{K}_{\text{shared}}(t) = \bigcap_{p_i \in [p]} \mathcal{K}_{p_i}(t)$$

where it is important to emphasise that \mathcal{K}_{p_i} are the uncommunicated knowledge states.

Example 2.9: Observe the following PEG played on a 3×3 grid with $A_p = \{p_1, p_2\}$ and $A_e = \{e\}$. Assume order-0 knowledge. We have the initial state

$$s_0 = \left(\begin{bmatrix} p_1 & 0 & 0 \\ 0 & 0 & e \\ p_2 & 0 & 0 \end{bmatrix}, p \right).$$

with the pursuers' turn to make a move. Following the first rule of communication we find that

$$\mathcal{K}_{p_1}^0(0) = \left(\begin{bmatrix} p_1 & 0 & 0 \\ 0 & * & * \\ p_2 & * & * \end{bmatrix}, p \right)$$

and

$$\mathcal{K}_{p_2}^0(0) = \left(\begin{bmatrix} p_1 & * & * \\ 0 & * & * \\ p_2 & 0 & 0 \end{bmatrix}, p \right).$$

However, if the pursuers are allowed to communicate through the second rule, their knowledge will instead be the intersection of the uncommunicated knowledge states

$$\mathcal{K}_{\text{shared}}(0) = \mathcal{K}_{p_1}^0(0) \cap \mathcal{K}_{p_2}^0(0) = \left\{ \left(\begin{bmatrix} p_1 & 0 & 0 \\ 0 & * & * \\ p_2 & 0 & 0 \end{bmatrix}, p \right) \right\}.$$

△

I. Knowledge Update Function

Having defined the knowledge representation for the PEG, it is now natural to seek an expression for the update function, in line with previous work by Gurov et al. in [5], that acts upon the knowledge representation whenever the pursuers or evaders make a move. A knowledge update is performed on the basis of the following three parameters: an old state (an old knowledge state), an action performed (a move as restricted by the movement relation) and a new observation (the observations made as defined by the visibility relation) and is given more generally by the mapping

$$\delta_{p_i}^k : \mathcal{K}_{p_i}^k \times \text{Mov}_{p_i} \times \mathcal{O}_{p_i} \rightarrow \mathcal{K}_{p_i}^k$$

where

$$\text{Mov}_{p_i} \subset E_m.$$

We will now define the knowledge update functions up to and including order two. The aim of the update function is

to determine the possible positions of every agent on the grid after either the pursuers or the evaders make a move. We define

$$\mathcal{P}_p^a(t) \subset V_n$$

to be the set of all possible positions for the agent $a \in A$ as perceived by the pursuer p at time point t . Furthermore, we define

$$\mathcal{P}_p(t) = \prod_{a \in A} \mathcal{P}_p^a(t) \subset \overbrace{V_n \times V_n \times \dots \times V_n}^{l+k \text{ times}}$$

to be the set of all possible positions for all agents on the grid as perceived by the pursuer p at time point t . The knowledge update function will now act upon the elements in $\mathcal{P}_p(t)$.

1) *Knowledge Update Function of Order Zero:* The knowledge update function of order zero is a function depending only on the new observation and is the same regardless if the pursuers or evaders moves

$$\mathcal{K}_p^0(t+1) = \delta_p^0(o_p) = o_p$$

Example 2.10: Observe the following pursuit-evasion game played on a 3×3 grid with $A_p = \{p\}$ and $A_e = \{e_1, e_2\}$. We have the initial state

$$s_0 = \left(\begin{bmatrix} e_2 & 0 & 0 \\ 0 & 0 & e_1 \\ 0 & p & 0 \end{bmatrix}, p \right)$$

and the knowledge state of 0-order

$$\mathcal{K}_p^0(0) = \left(\begin{bmatrix} * & 0 & * \\ * & 0 & * \\ 0 & p & 0 \end{bmatrix}, p \right).$$

The pursuer moves into the leftmost column

$$s_1 = \left(\begin{bmatrix} e_2 & 0 & 0 \\ 0 & 0 & e_1 \\ p & \leftarrow & 0 \end{bmatrix}, e \right)$$

where it now makes the following observation

$$o_p = \left\{ \left(\begin{bmatrix} e_2 & * & * \\ 0 & * & * \\ p & 0 & 0 \end{bmatrix}, e \right) \right\}.$$

The knowledge updates as follows:

$$\mathcal{K}_p^0(1) = \{s_1, s'_1, s''_1, s'''_1\} = \left\{ \left(\begin{bmatrix} e_2 & * & * \\ 0 & * & * \\ p & 0 & 0 \end{bmatrix}, e \right) \right\}$$

△

2) *Knowledge Update Function of Order One:* The knowledge update function of order one is not necessarily only a function depending on the new observation. It is also a function of the previous knowledge state. The knowledge update is different depending on the turn. One case where order-0 knowledge and order-1 knowledge coincide is when the history of the game $|\mathcal{H}| = 1$.

Pursuers' turn: Let us determine the possible positions

$$\mathcal{P}_p(t+1)$$

of every agent on the grid in the perspective of pursuer p after making a move such that

$$(pos_p(t), pos_p(t+1)) \in E_m.$$

If another pursuer p_i is visible to the pursuer p , namely

$$(pos_p(t+1), pos_{p_i}(t+1)) \in E_v$$

$\mathcal{P}_p^{p_i}(t+1)$ collapses to a singleton set

$$\mathcal{P}_p^{p_i}(t+1) = \{pos_{p_i}(t+1)\}$$

If this is not the case, the possible positions $\mathcal{P}_p^{p_i}(t+1)$ for a pursuer p_i is given by:

$$\begin{aligned} \mathcal{P}_p^{p_i}(t+1) = & \{loc : \forall loc_{p_i} \in \mathcal{P}_p^{p_i}(t) (loc_{p_i}, loc) \in E_m\} \\ & \setminus \{loc : (pos_p(t+1), loc) \in E_v\} \end{aligned}$$

Similarly, if an evader e_i is visible to the pursuer p , that is

$$(pos_p(t+1), pos_{e_i}(t+1)) \in E_v$$

the set $\mathcal{P}_p^{e_i}(t+1)$ collapses to a singleton set. Otherwise, the possible positions $\mathcal{P}_p^{e_i}(t+1)$ for an evader e_i is given by:

$$\mathcal{P}_p^{e_i}(t+1) = \mathcal{P}_p^{e_i}(t) \setminus \{loc : (pos_p(t+1), loc) \in E_v\}$$

Finally, since pursuers and evader cannot occupy the same locations, we define the pruned set:

$$\begin{aligned} \mathcal{P}_p^{\text{pruned}}(t+1) = & \mathcal{P}_p(t+1) \setminus \{C : C \in \mathcal{P}_p(t+1) \text{ and} \\ & \exists e_i \in A_e, p_j \in A_p \text{ s.t. } pos_{e_i}(t+1) = pos_{p_j}(t+1)\} \end{aligned}$$

We can now formally define the knowledge update function as follows:

$$\mathcal{K}_p^1(t+1) = \delta_p^1(\mathcal{K}_p^1(t), o_p) = \bigcap_{p_i \in [p]} \{(C, e) : C \in \mathcal{P}_p^{\text{pruned}}(t+1)\}$$

Notice that if we assume the first communication rule we get:

$$\mathcal{K}_p^1(t+1) = \{(C, e) : C \in \mathcal{P}_p^{\text{pruned}}(t+1)\}$$

If we on the other hand assume visibility-based communication, the updated knowledge state would consist of the intersection of the knowledge states of all communicating pursuers.

Example 2.11: Assume the following game played on a 4×4 grid with $A_p = \{p_1, p_2\}$ and $A_e = \{e_1\}$.

$$s_0 = \left(\begin{bmatrix} 0 & 0 & 0 & 0 \\ p_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ p_1 & 0 & e_1 & 0 \end{bmatrix}, p \right)$$

with the pursuers making the following move such that

$$s_1 = \left(\begin{bmatrix} 0 & 0 & 0 & 0 \\ \downarrow & 0 & 0 & 0 \\ p_2 & 0 & 0 & 0 \\ \rightarrow & p_1 & e_1 & 0 \end{bmatrix}, e \right)$$

Since p_1 knows the number of agents in the game and can observe every agent using the visibility relation, the order-1

knowledge of p_1 in state s_0 is a singleton set comprised of the following element:

$$\mathcal{K}_1^1(0) = \{s_0\} = \left\{ \left(\begin{bmatrix} 0 & 0 & 0 & 0 \\ p_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ p_1 & 0 & e_1 & 0 \end{bmatrix}, p \right) \right\}$$

However, after making a move towards the evader the pursuer no longer observes the other pursuer and can therefore deduce that p_2 has either moved up or down. The first pursuers knowledge is no longer a singleton set and is comprised of the unordered pair:

$$\mathcal{K}_1^1(1) = \{s_1, s'_1\} = \left\{ \left(\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ p_2 & 0 & 0 & 0 \\ 0 & p_1 & e_1 & 0 \end{bmatrix}, e \right), \left(\begin{bmatrix} p_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & p_1 & e_1 & 0 \end{bmatrix}, e \right) \right\}$$

△

Evaders' turn: Following the same reasoning as before, let us determine $\mathcal{P}_p(t+1)$ with respect to pursuer p . Firstly, every evader e_i will make a move such that:

$$(pos_{e_i}(t), pos_{e_i}(t+1)) \in E_m$$

If an evader e_i is visible to the pursuer p , namely

$$(pos_p(t+1), pos_{e_i}(t+1)) \in E_v$$

the set $\mathcal{P}_p^{e_i}(t+1)$ collapses to a singleton set given by:

$$\mathcal{P}_p^{e_i}(t+1) = \{pos_{e_i}(t+1)\}$$

If this is not the case, the possible positions $\mathcal{P}_p^{e_i}(t+1)$ for an evader e_i is

$$\mathcal{P}_p^{e_i}(t+1) = \{loc : \forall loc_{e_i} \in \mathcal{P}_p^{e_i}(t) (loc_{e_i}, loc) \in E_m\} \setminus \{loc : (pos_p, loc) \in E_v\}$$

Let us now define the pruned set

$$\mathcal{P}_p^{\text{pruned}}(t+1) = \mathcal{P}_p(t+1) \setminus \{C : C \in \mathcal{P}_p(t+1) \text{ and } \exists e_i \in A_e, p_j \in A_p \text{ s.t. } pos_{e_i}(t+1) = pos_{p_j}(t+1)\}$$

which takes into account that no evader can occupy the same positions as a pursuer. Like before, we can now define the knowledge update function when the evaders make a move as follow:

$$\mathcal{K}_p^1(t+1) = \delta_p^1(\mathcal{K}_p^1(t), o_p) = \bigcap_{p_i \in [p]} \{(C, e) : C \in \mathcal{P}_p^{\text{pruned}}\}$$

Example 2.12: Assume the following game played on a 3×3 grid with $A_p = \{p_1, p_2\}$ and $A_e = \{e\}$ with the initial state:

$$s_0 = \left(\begin{bmatrix} p_1 & 0 & 0 \\ p_2 & 0 & e \\ 0 & 0 & 0 \end{bmatrix}, e \right)$$

The evader proceed to make a move such that

$$s_1 = \left(\begin{bmatrix} p_1 & 0 & e \\ p_2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, p \right)$$

where the knowledge of the pursuers differ depending on the rule of communication. Firstly, assume no communication. We then have that the knowledge of p_1 is a singleton set

$$\mathcal{K}_{p_1}^1(1) = \{s_1\}$$

and the knowledge of p_2 is the unordered pair:

$$\mathcal{K}_{p_2}^1(1) = \{s_1, s'_1\} = \left\{ \left(\begin{bmatrix} p_1 & 0 & 0 \\ p_2 & 0 & 0 \\ 0 & 0 & e \end{bmatrix}, p \right), \left(\begin{bmatrix} p_1 & 0 & e \\ p_2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, p \right) \right\}$$

However, if we follow the second rule of communication we have that

$$(p_1, p_2) \in E_c \iff p_1 \in [p_2]$$

which implies that:

$$\mathcal{K}_{p_1}^1(1) = \mathcal{K}_{p_2}^1(1) = \{s_1\} \cap \{s_1, s'_1\} = \{s_1\}$$

△

3) *Knowledge Update function of Order Two:* Defining a knowledge update function of order two with first communication rule is extremely cumbersome and will be disregarded in this thesis. However, when there is communication, we can define the update function in the following way. The second order knowledge of pursuer p is defined to be

$$\mathcal{K}_p^2 = \{K_p^1, \mathbb{K}_{\text{all other pursuers}}^1\}$$

where

$$\mathbb{K}_{p_i}^1 = \mathcal{K}_p^1 \cap \mathcal{K}_{p_i}^1$$

whenever

$$(p, p_i) \in E_c$$

and

$$\mathbb{K}_{p_i}^1 = \emptyset$$

otherwise.

J. Knowledge-Based Strategies of different order

When a multi-player game like the PEG has been defined, the question of how agents might achieve a certain objective naturally arises. An algorithm an agent might use to achieve an objective is called a strategy. In its most simple form, a strategy for a pursuer p is a mapping

$$S : \mathcal{K}_p \rightarrow Act_p$$

from knowledge states to actions where

$$Act_p \in \text{Mov}_p$$

is an element of the possible moves a pursuer might make. Strategies of particular importance in this thesis are known as knowledge-based strategies [5] and consists of the following:

- 1) A knowledge representation that contains the current knowledge of an agent.
- 2) An action mapping. A function that maps a pursuers current knowledge to a prescribed action.
- 3) A knowledge update function. A function that updates the current knowledge for the pursuers after every transition.

We define an order- k knowledge-based strategy for a pursuer p to be a mapping from knowledge states to actions that uses an order- k knowledge representation together with an order- k knowledge update function. That is,

$$S_p^k : \mathcal{K}_p^k \rightarrow \text{Act}_p.$$

1) *Knowledge-based strategy of order zero:* Using the previously defined knowledge representation and knowledge-update function of order zero, the most natural way to assign an action to the pursuers is to move towards the closest evader whenever the pursuer observes an evader. Otherwise, move randomly.

Definition 2 (Strategy S^0): The pursuer's strategy is to move towards the closest visible evader, where closest is defined by the metric given in the definition of radius based visibility. If there are no visible evaders, pursuers move randomly. \triangle

2) *Knowledge-based strategy of order one:* Following in the footsteps of Goobar and Söderberg, we define a knowledge-based strategy of order one to be a strategy where pursuers chooses the move that minimises the uncertainty of the positions of the evaders. This is a strategy Goobar and Söderberg [2] referred to as the 'Removing Ones' strategy. One can ask oneself about the rationale behind a strategy where the goal is to remove uncertainty. Think about it like this - by eliminating the unknown bit by bit, you shrink down the possible locations of the evaders. This can be seen as a pincer movement, where the possible positions are reduced until there is only certain positions left. Instead of moving randomly, uncertainty is purposefully reduced until the evaders are caught. Let us now define such a strategy theoretically. We define all possible evader states, \mathbb{E}_p , for a pursuer p as follows

$$\mathbb{E}_p = \prod_{e \in A_e} \mathcal{P}_p^e.$$

We may now formally define a first order strategy. This is a redefinition of a strategy defined by Goobar and Söderberg in [2].

Definition 3 (Strategy S_1 (called S_3 in [2])):

The pursuers objective is to minimise the individual uncertainty regarding the positions of the evaders. Every pursuer p chooses an action such that

$$|\mathbb{E}_p(t+1)|$$

is minimised in the worst case scenario. If a certain pursuer p has knowledge about the current positions of one or more evaders it will follow the logic of S_0 . \triangle

We need to explain what we mean by the worst case scenario. Since pursuers does not know beforehand exactly what they will see after making a move, it is impossible to predict the size of

$$|\mathbb{E}_p(t+1)|.$$

Nevertheless, if we assume that no evaders will be visible to the pursuer after making a move, we can still calculate the move or moves that will decrease the uncertainty regarding the positions of the evaders the most.

Example 2.13: Observe the following initial state s_0 of a PEG played on a 4×4 grid with $A_p = \{p_1, p_2\}$ and $A_e = \{e_1, e_2\}$. Assume there is order-1 knowledge and we follow the second rule of communication. We have that

$$s_0 = \left(\begin{bmatrix} 0 & 0 & p_1 & 0 \\ 0 & e_1 & 0 & 0 \\ 0 & 0 & p_2 & 0 \\ 0 & e_2 & 0 & 0 \end{bmatrix}, p \right)$$

and since $p_1 \in [p_2]$ we have that

$$\begin{aligned} \mathcal{K}_{p_1}^1(0) &= \mathcal{K}_{p_2}^1(0) = o_{p_1} \cap o_{p_2} \\ &= \left(\begin{bmatrix} 0 & 0 & p_1 & 0 \\ * & * & 0 & * \\ 0 & 0 & p_2 & 0 \\ * & * & 0 & * \end{bmatrix}, p \right) \end{aligned}$$

Each pursuer will now choose a move with the purpose of minimising the individual uncertainty regarding the position of the evader the most. This is in this case equivalent to removing as many * as possible. Consequently, pursuer p_1 will move down, and pursuer p_2 will have the option of moving up or down. Both moves are for the pursuer equally advantageous. Following is one possible subsequent state:

$$s_1 = \left(\begin{bmatrix} 0 & 0 & \downarrow & 0 \\ 0 & e_1 & p_1, p_2 & 0 \\ 0 & 0 & \uparrow & 0 \\ 0 & e_2 & 0 & 0 \end{bmatrix}, e \right)$$

\triangle

3) *Knowledge-based strategy of order two:* The following example will give the intuition behind the formulation of our second order strategy.

Example 2.14: Let us return to example 2.13 with the initial state

$$s_0 = \left(\begin{bmatrix} 0 & 0 & p_1 & 0 \\ 0 & e_1 & 0 & 0 \\ 0 & 0 & p_2 & 0 \\ 0 & e_2 & 0 & 0 \end{bmatrix}, p \right).$$

It would be intuitively much more beneficial for the pursuers to make the following coalition move to the state

$$s_1 = \left(\begin{bmatrix} 0 & 0 & \downarrow & 0 \\ 0 & e_1 & p_2 & 0 \\ 0 & 0 & \downarrow & 0 \\ 0 & e_2 & p_2 & 0 \end{bmatrix}, e \right)$$

where every remaining evader is now located by the set of pursuers. \triangle

Definition 4 (Strategy S_2):

The pursuers objective is to minimise the collective uncertainty regarding the locations of the evaders. For a certain pursuer $p \in A_p$, if

$$|[p]| = 1$$

it will follow the strategy S_1 . However, if

$$|[p]| \neq 1$$

the pursuer p chooses an action such that

$$\left| \bigcap_{p_i \in [p]} \mathbb{E}_{p_i}(t+1) \right|$$

is minimised in the worst case scenario. Similar to strategy S_1 , if a certain pursuer p has knowledge about the current positions of one or more evaders, S_2 will fall back to S_0 \triangle

III. SIMULATION SETUP

Having concluded that the knowledge based strategies applied in [2] were all of order-1 knowledge (or less), we wanted to develop a strategy of order-2 knowledge and evaluate its performance, compared to the strategies proposed by Goobar and Söderberg in [2]. To do this evaluation, we have used the simulation environment developed in [2]. This is an object oriented Python written script. However, some changes to the code were needed to be made made to adapt it to the newly written strategy. The implemented order-2 knowledge strategy is outlined in Definition 4 and a pseudo code description of its implementation can be seen below.

All code can be found in the Github repository for this project: https://gits-15.sys.kth.se/jacobgra/kex_O1a

The strategies that will be analysed are the following

- S_{opt} - the perfect strategy. This strategy simulates a game with perfect information. The pursuers always move to the closest evader. This strategy works as a reference for the performance of other strategies to compare with.
- S_{rand} - random movement
- S_0 - move towards closest visible evader, else random movement
- S_1 - Removing Ones Individually with second rule of communication
- S'_1 - Removing Ones Individually with first rule of communication
- S_2 - Removing Ones Commonly

To obtain a needed number of iterations per simulation setup an RSEM-test¹ was conducted. This was done to obtain higher validity in our results. To maintain relevancy with respect to Goobar and Söderberg's result, we also determined the efficiency of our strategy using the same criteria they used.

- 1) Average time to capture all evaders as a function of the grid-size.
- 2) Average time to capture all evaders as a function of the number of evaders.
- 3) Average time to capture all evaders as a function of the number of pursuers.

IV. RESULTS

As we can see in Figure 1, the number of iterations which will result in an RSEM of less than 1% is 9000. Hence, all following simulations have been run with 9000 of iterations per setup, to obtain a high validity of the results.

In Figure 2 we can observe the result dependency on the grid size. As can be seen, with the given parameters, our new

¹Relative Standard Error of the Mean

Algorithm 1: Removing Ones Collectively Strategy

Data: Positions of pursuers and evaders, Knowledge matrices, Legal Moves

Result: Designated moves for pursuers for the current turn

$M \leftarrow$ array of designated moves;

$S \leftarrow$ list of sets;

for each pursuer do

if pursuer sees all evaders then

 find move towards closest evader ; /* old algorithm */

 add move to M ;

end

end

for each pursuer do

if pursuer sees another pursuer then

 add pair of pursuers that see each other S ;

end

end

while no disjoint sets in S do

for set in S do

for set in S do

if sets contain common elements then

 merge sets

end

end

end

end

for set in S do

 check all combinations of movement for a set of pursuers that see each other to establish which movement reduces the uncertainty the most; add these moves for each "coalition" of pursuers to M ; /* note that a coalition can consist of 1 pursuer */

end

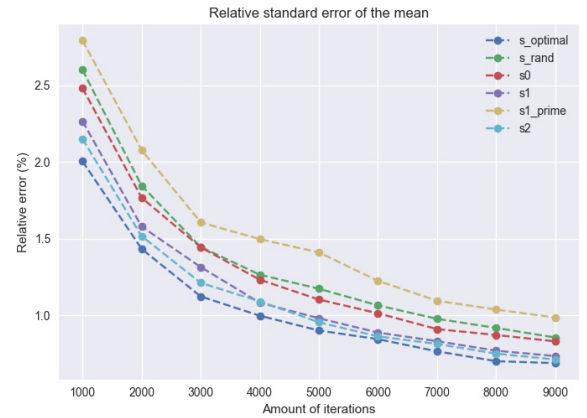


Fig. 1. RSEM-graph. Simulation parameters: $n = 5$, $|A_p| = 4$, $|A_e| = 2$

strategy does not perform differently from either S_1 or S_0 . All strategies perform better than random movement, but are

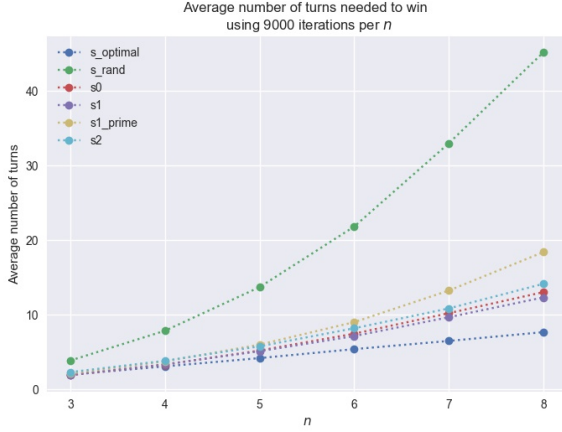


Fig. 2. Grid-size dependence. Simulation parameters: $|A_p| = 4$, $|A_e| = 2$

outperformed by the perfect strategy. When the grid size increases we observe no changes in inter-strategic performance, but an overall increase in the number of turn needed to capture all evaders. The communications-less strategy, S'_1 performs in between the random movement and the other strategies.

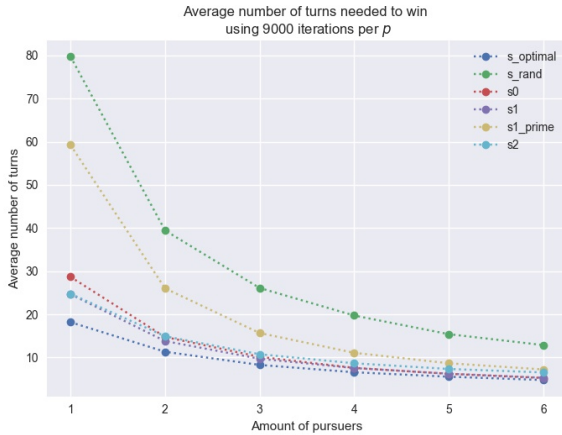


Fig. 3. Pursuer dependence. Simulation parameters: $|A_e| = 2$, $n = 5$

If we study the impact of the number of pursuers on the turns required to capture all evaders, the pattern from the Figure 2 persists. In Figure 3 it is almost impossible to discern the data points representing the strategies S_1 and S_2 . What we perhaps observe is actually a slight under performance of the strategy S_2 . We also observe that the communications-less strategy, S'_1 again performs in between the random movement and the other strategies.

The same result as in Figure 3 holds for the simulation for evader dependence and can be seen in Figure 4.

As was already concluded in [2], the knowledge based strategies outperform the random movement "strategy". This can be seen in Figures 2, 3 and 4. However, the second order knowledge based strategy, S_2 devised in this thesis does not seem to perform better than the first order knowledge based strategy S_1 . In figure 2 we can see that as the amount of pursuers and evaders are held constant ($|A_p| = 4$, $|A_e| = 2$)

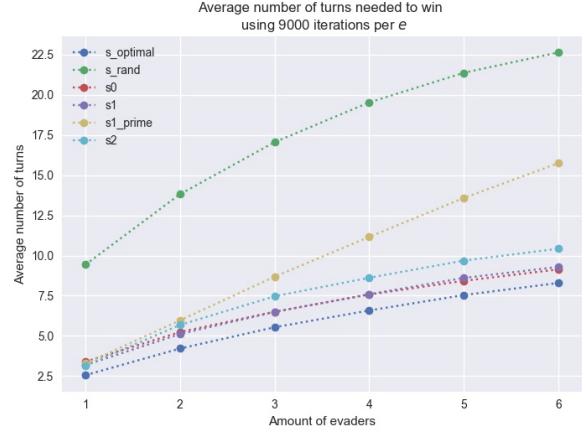


Fig. 4. Evader dependence. Simulation parameters: $|A_p| = 4$, $n = 5$

and we increase the grid size, no difference can be discerned between the result of the strategies.

V. DISCUSSION

Concerning our formalisation of the PEG and our simulation, multiple things have arisen that are interesting to discuss.

1) *Formalisation:* The formalisation of the PEG that was developed in this thesis gives ample opportunities for variation. By clearly defining concepts as visibility, movement and communication, as well as knowledge and knowledge updates one achieves a good foundation to explore and vary parameters within the game. With this construction that we have detailed in this thesis it is also close at hand to translate the results to a more generalised group of games as proposed by Gurov et al. [5].

2) *Simulation:* The results obtained from the simulations tell several interesting stories. Firstly, we can see that knowledge based strategies outperform random movement. The worst performance in all simulations is always the random movement strategy. Secondly, we observe that there is a quite large discrepancy between strategies which allow for communication compared to communication-less strategies. In fact, our first order knowledge based communication-less strategy S'_1 performs worst of all studied knowledge-based strategies. One might ponder upon why a designated movement towards removing uncertainty would perform worse than prescribing random movement, when a pursuer does not observe any evaders. Earlier, we gave a rationale for the opposite, which clearly did not hold. Perhaps, the movement towards the uncertain makes the pursuer move purposefully away from the evader when it no longer spots them, since the knowledge they actually have will be surrounding the evader's old position. Therefore, this strategy actually performs actively worse than the corresponding strategy S_0 which moves a pursuer purposefully towards an evader if it sees one, or else moves randomly. Thirdly, the results of communication-based strategies of order-0, order-1 and order-2 knowledge are practically indiscernible from one another, if anything S_2 of order-2 knowledge is actually performing worse than S_1 and S_0 . This could suggest that the possible gains from

higher order knowledge based strategies in this game might be limited. However, the defined second-order strategy is only one of many possible strategies, and therefore no decisive conclusions can be drawn about the general performance of higher order knowledge based strategies in PEG:s.

VI. CONCLUSION

We have seen that PEG:s and knowledge representations can be neatly formalised mathematically. Since our definitions are relation-based, it is possible to easily translate our definition of the PEG to games played on arbitrary maps, not just finite grids. In our simulation, we can see that knowledge-based strategies outperform strategies solely based on randomness. However, strategies based upon minimising the uncertainty of the evaders' positions perform worse than observation-based strategies. Lastly, higher order knowledge based strategies do not necessarily perform better than strategies of lower order.

VII. FUTURE WORK

There is a myriad of questions within the field of PEG:s to investigate in the future. For instance, it would be interesting to explore how one might simulate PEG:s and knowledge-based strategies on arbitrary maps. Namely, write a program that takes a number of locations, movement relations and visibility relations as inputs and simulates the induced PEG. How would one implement first order knowledge on arbitrary maps?

If we instead were to define the evaders to be rational agents, you could also try to construct knowledge based strategies for the evaders in order to aid them in avoiding the pursuers. What if all agents $a \in \mathcal{A}$ are rational?

When working with this thesis, we initially intended to apply the research results of our supervisor Dilian Gurov to PEG:s. Namely, transforming the PEG into a MAGIIAN [5] (Multi Agent Game of Imperfect Information Against Nature) and apply a construction known as a MKBSC (Multiplayer Knowledge Based Subset Construction) on the MAGIIAN in order to extract better performing strategies. However, due to the large amount of possible states of the PEG it is not very easily adaptable to the MKBSC algorithm which works better on games with fewer possible game states. Therefore, we quickly deemed it to difficult to continue to pursue work in this direction. Nevertheless, The translation of the developed formalisation into a MAGIIAN would be an interesting task as well as a sanity check that the formalisation yields reasonable results also within this more general realm of games.

ACKNOWLEDGEMENT

The authors would like to thank the supervisor Dilian Gurov for his support, valuable inputs and optimism during the work with this thesis.

REFERENCES

- [1] 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, July 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>
- [2] T. Gabi Goobar and S. Söderberg, "Knowledge based strategies in grid-based pursuit-evasion games of imperfect information," 2021.
- [3] L. Doyen and J.-F. Raskin, "Games with imperfect information: theory and algorithms." *Lectures in Game Theory for Computer Scientists*, vol. 10, 2011.
- [4] D. Gurov, V. Goranko, and E. Lundberg, "Knowledge-Based Strategies for Multi-Player Games with Imperfect Information," *Slides from ZTSrIO Seminar*, 2021.
- [5] —, "Knowledge-based strategies for multi-agent teams playing against nature," *Artificial Intelligence*, p. 103728, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0004370222000686>

Strategy Synthesis for Multi-agent Games of Imperfect Information With Partially Given Strategies

Oden Allen and Erik Skog

Abstract—Finding strategies for games have been of interest to humans throughout history. With the advancement of technology and the way the financial market is compounded, enormous time and resources are spent on modelling real world problems as games and searching for strategies modelled to enhance productivity and rule out inefficiencies. This thesis aims to investigate the existence of strategies that would allow players (agents) to complete common objectives when one category of agents already have a given strategy. This is done through studying an example and investigating the application and implication of the introduction of an abstraction function. The performed study concluded that if such a function could be more rigorously mathematically formulated, it could increase the effectiveness of strategy searches and synthesis in the field.

Sammanfattning—Människor har alltid varit intresserade av att hitta strategier för spel. I och med teknikens utveckling och finansmarknadens uppbyggnad läggs enorm tid och resurser på att modellera verkliga problem som spel och söka efter strategier för att öka produktiviteten och minska ineffektivitet. Syftet med rapporten är att undersöka om det finns strategier som gör det möjligt för spelarna (agenterna) att uppnå gemensamma mål när en kategori av agenter redan har en given strategi. Detta görs genom att studera ett exempel och undersöka tillämpningar och konsekvenserna av att införa en abstraktionsfunktion. I studien drogs slutsatsen att om en sådan funktion kunde formuleras strikt matematiskt skulle den kunna öka effektiviteten i strategisökningar inom området.

Index Terms—Strategy synthesis, MAGIIAN, Imperfect information, Abstraction function, MAGSIIAN.

Supervisors: Dillian Gurov

TRITA number: TRITA-EECS-EX-2022:180

I. INTRODUCTION

If one is to consider a game, where two or more people work together to achieve some common objective, for example winning in bridge, it is a fair assumption to make that at least one player has a strategy. A question then arises, can the player or players without strategy deduce which actions to take to achieve the common objective of the game? This thesis will treat the subject of multiplayer games with two categories of players, one category where the group has pre-defined strategies and the other category, where the player lack strategies. The strategies of the former group are known to all players in the game. This thesis will investigate if a winning strategy can be synthesized for the players lacking strategies. This is done by creating a new game, a MAGSIIAN, by abstracting away the category of players whose strategies

are known to all. In this new game a construction called multi-agent knowledge based subset construction (MKBSC), as described in [1], is used onto the MAGIIAN to search for winning strategies and then transform them back to the original game.

A. Objective

In this thesis we aim to investigate strategy synthesis for a game of imperfect information of two categories of players (from here on, agents), one category where the agents have pre-defined strategies (PDS), that informs what each agent should do at what location of the game, and one category of agents where no such strategies exist and must be searched for, these are called NPDS agents. This game is played with imperfect information for the NPDS agents, later defined in III-B, and who's strategies must be searched for. The search for strategies will be conducted with a game abstraction of the original game.

B. Approach

We are approaching the problem with the base and mathematical formulation found in [1] and [2]. We will present a mathematical abstraction from the MAGSIIAN III-B, to a MAGIIAN [1]. The pre-defined strategies of the PDS agents are considered common knowledge, thus making the actions taken by the PDS agents known throughout the game. If one changes perspective, the agents with PDS can be viewed as an action taken by nature, this is possible because of their pre-determined actions. Nature can be viewed as an opposing player or players taking actions, who's moves are considered non-deterministic due to the agents not having knowledge regarding their strategy. If no decision is needed by one of the agents, then that agents' move is out of my control and thus can be abstracted away into nature. When this perspective is taken, the MAGSIIAN game can be viewed as a MAGIIAN. If a strategy is found we will introduce a strategy translation, which will be discussed in the example in section IV.

C. Delimitation's

The defined MAGSIIAN is based on a frame of rules defined later in III-B, additionally the game also assumes the following:

- The agents do not communicate during the course of the game and can be looked at as a modified version of the

(YN) case in [3] where without pre-defined strategies where NPDS have knowledge of agents with predefined strategies but not the other way around.

- The PDS agents, defined in the MAGSIAN III.1, can be viewed as an opponent since its behavior is fixed by its strategies, but in this thesis we will only investigate the case where the two categories of agents are working together as a team.

The scope of this thesis is limited to only investigate strategies of finite-memory. Mathematical formulations and definitions are not rigorously proven, instead they are claims that are based on logic, deduction and intuition extrapolated from knowledge of the field.

II. BACKGROUND

This section of the thesis will introduce concepts that are crucial for the understanding of the proposed ideas and functions. It will start by introducing the most basic concept of **single player games** where **perfect** and **imperfect** information will be introduced and explained. From single player games we will move into **multiplayer games**. After the different game types have been explained, we will go into depth explaining inherent concepts and constructions of these games, they are: **objectives**, **strategies**, single player and multiplayer **constructions** and finally ending with the explanation of **transducers**.

A. Single player game with perfect information vs nature

Consider the game of Tic-Tac-Toe, this game can be viewed as a single player game, where one player is playing vs an opponent, here after referred to as nature. This game can be viewed as a game graph tuple $G = \{L, l_0, \Sigma, \Delta\}$. Where L are all the possible variations of the game board, referenced as locations. l_0 is the initial configuration of the board, Σ is a finite set of actions the player can perform and Δ are the transitions from one variation of the board to another. Formally defined as:

Definition II.1 (Game with Perfect information). Let G be a game graph tuple consisting of,

$$G = \{L, l_0, \Sigma, \Delta\}$$

where:

- L is a finite set of **locations** describing the configurations of the game.
- $l_0 \in L$ is the **initial location**.
- Σ is a finite set of actions available for the player.
- $\Delta \subseteq L \times \Sigma \times L$ are the **transitions**, formulating all the edges between all locations in the game.

The course of the game can be described by the following rules:

- 1) Player one performs one action $\sigma_i \in \Sigma$ on a location $l_i \in L$.
- 2) Nature resolves the non-determinism, with one transition $(l_i, \sigma_i, l'_i) \in \Delta$.
- 3) the new location is $l'_i \in L$

A single player game has perfect information if the player can observe all locations of the game, as previously defined

by [4]. An example of a game with perfect information can be seen in fig. 1. Illustrating this concept using the game Tic-

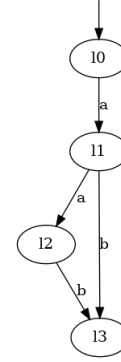


Fig. 1. Single player game of perfect information. The agent is able to differentiate between all locations of the game.

Tac-Toe the first location in fig.1, l_0 , is the blank board. The first move, $\sigma_0 \in \Sigma$, done by player 1 is putting a cross in the center tile. Next, player two, or nature, places a circle in any available tile which chooses the next location. Now that it's player one's turn and the game has reached a new location l_1 and a transition $(l_0, \sigma_0, l_1) \subseteq \Delta$ has taken place. If one only considers player one's actions, player two only produces a new configuration of the board to be acted upon by player one and one does not have to take it into consideration.

B. Single Player Game With Imperfect information

Using the game defined in II-A, we can view a single player game of imperfect information as the following.

Definition II.2 (Game with imperfect information against nature (GIHAN)). Let a single player game with imperfect information G be seen as a tuple:

$$G = \{L, l_0, \Sigma, \Delta, \mathcal{O}\}$$

where, L, l_0, Σ and Δ are defined as in II-A.

- \mathcal{O} is a set of **observations** where \mathcal{O} partitions L as $\mathcal{O} \subseteq 2^L$.

An observation, $o_i \in \mathcal{O}$ are the states where the player is unable to distinguish between sets of locations.

This inability to distinguish different locations in the form of observations are the inherent properties of games with imperfect information, this have been defined in [5]. An example of a game with imperfect information can be seen in fig. 2.

If every observation, $o_i \in \mathcal{O}$, only contains a singleton, the game is considered to have perfect information, since the single set in each of the observations is a location.

C. MAGSIAN

A single-player game with imperfect information, defined in II-B, can be generalized into a multiple agent game with imperfect information (MAGSIAN) following the definition in [1]. The definition is as follows:

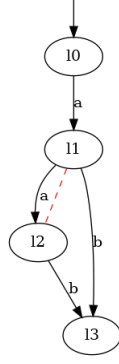


Fig. 2. Single player game of imperfect information, where the red dotted line illustrates the locations the agent is unable to differentiate between, making $\mathcal{O}_1 = \{l_1, l_2\}$.

Definition II.3 (Multi-agent game with imperfect information against nature (MAGSIAN)). Let a single-player game with imperfect information be a tuple $G = \{L, l_o, \Sigma, \Delta, \mathcal{O}\}$. Let G' be the combined game for n agents.

$$G' = \{Agents, L, l_o, \Sigma, \Delta, \mathcal{O}\}$$

Where:

- i $Agents = \{agt_1, agt_2, \dots, agt_n\}$ are the **agents** playing vs Nature.
- ii L is a finite set of **locations** describing the configurations of the game.
- iii $l_o \in L$ is the **initial location**.
- iv $\Sigma = \Sigma_1 \times \dots \times \Sigma_n$ are the **action profiles** of the team.
- v $\Delta \subseteq L \times \Sigma \times L$ are the **transitions**, formulating all the edges between all locations in the game.
- vi $\mathcal{O} = \mathcal{O}_1 \times \dots \times \mathcal{O}_n$ are the **observation profiles** of the agents.

This results in a multi-agent game with imperfect information, although all the agents does not necessarily have imperfect information.

D. Objectives

1) *Play*: A full play is all the moves made during the course of an ongoing game until the end, and can be described as a series of alternating locations and actions for an agent $\pi = l_0 \sigma_1 l_1 \sigma_2 l_2 \dots$. A full history are all the moves that were made during the game and consists of an alternating, finite series of locations and actions $\pi(i) = l_0 \sigma_1 l_1 \sigma_2 l_2 \dots l_i \sigma_i$. Both a play and a history are variations of full play and full history. Both play and history have the action component σ removed from full play and full history. Thus, they are defined as $\pi = l_0 l_1 \dots$ for a play and $\pi(i) = l_0 l_1 \dots l_i$ for a play.

2) *Reachability objective*: A **reachability objective** can be defined by a non-empty set of locations $\mathcal{R} \subseteq L$.

A play $\pi = l_0 l_1 l_2 \dots$ is winning if it visits some location in \mathcal{R} .

$$\mathcal{R} \subseteq L$$

$$\mathcal{R} = \{\pi = l_0 l_1 l_2 \dots \mid \exists l_i \in \mathcal{R}, i \geq 0\}$$

A reachability objective is observable for an $agent_i$, if it is a union of observations observed by $agent_i$ $\mathcal{R}_1 \cup \mathcal{R}_2 \cup \dots \cup \mathcal{R}_i$, and can therefore be defined alternatively as a set $\mathcal{R} \subseteq \mathcal{O}$

3) *Safety objective*: A safety objective is a set of locations that the agent have to visit for every play in order to be winning [1].

$$\mathcal{S} \subseteq L$$

$$\mathcal{S} = \{\pi = l_0 l_1 l_2 \dots \mid \forall l_i \in \mathcal{S}, i \geq 0\}$$

E. Strategies

A strategy can be described as a function that has locations or observations as input and returns an action, either based on what is currently observed and/or for all previous observations.

1) *Memory-less Strategy*: When a strategy makes decisions based solely on what is currently observed, it is called a memory-less strategy, since it does not take into consideration previous observations. It can thus be formalized as

$$\alpha_i : \mathcal{O}_i \rightarrow \Sigma_i$$

where $o_i \in \mathcal{O}$ is the current observation being made, and Σ the actions that can be taken.

2) *Perfect-recall strategy*: There also exists a type of strategy called perfect-recall where the strategy takes into account all the previous observations in the play, $\pi = \{o_1 o_2 \dots o_i, \forall o \in \mathcal{O}\}$, and the currently observed game state and can be formalized as:

$$\alpha_i : \mathcal{O}_i^+ \rightarrow \Sigma_i$$

where \mathcal{O}^+ are all the previous observations and Σ is the actions that can be taken.

3) *Finite-memory Strategy*: A finite-memory strategy is commonly represented by the memory-states M and is can be modelled by a transducer. This will be further explained in definition II-H. Below is the formalization for the general finite-memory strategy.

$$\alpha_i : M_i \rightarrow \Sigma_i$$

F. Knowledge-based subset construction

The Knowledge-Based Subset Construction (KBSC) is a mathematical construction used to transform a single player game of imperfect information into a single player game of perfect information by reasoning about the knowledge of the agent [4]. This is done by making an expansion of the original game G^K . This expansion is defined as

Definition II.4 (Knowledge-based subset construction (KBSC)). Let G be a GIIN and let G^K be the expansion after applying the KBSC onto G then G^K is defined as

$$G^K = \{S, s_0, \Sigma, \Delta^k\}$$

where

- 1) S are the **knowledge states** and is the power set of L excluding the empty set, meaning it is all the possible combinations of elements in the set L except the empty set: $S = 2^L / \emptyset$. Raskin et al. [4] describes the knowledge

states as "Each state in G^K is a set of states of G which represents the knowledge of Player 1".

- 2) s_0 is the **initial memory state** of the expanded game.
- 3) Σ is the **action profile** to the agent.
- 4) $\Delta^k \subseteq S \times \Sigma \times S$ are the **transitions** between states where $s, s' \in S, \sigma \in \Sigma$ such that $(s, \sigma, s') \subseteq \Delta^k$

It was shown in [4] that a winning memory-less strategy found in the expanded game can be transformed into a finite-memory strategy in the original game in the form of a transducer, which is further explained in definition II.6. This is due to the property of strategy preservation meaning that for a winning reachability objective \mathcal{R} in G it is represented in G^K as \mathcal{R}^k . The winning strategies used to reach \mathcal{R}^k can be translated into winning strategies to reach \mathcal{R} . This is advantageous due to the fact that it is easier to search for memoryless strategies in a game of perfect information than in a game of imperfect information [4]. This is due to the lower computation cost of finding a memory-less strategy compared to a finite-memory strategy, and which can be more easily found in a perfect information game than in an imperfect information game. If a proof holds for memory-less strategies, it also holds for finite-memory and perfect-recall strategies [3].

G. Multi-agent Knowledge Based Subset Construction

The KBSC can be generalized from GIIN, as defined in definition II.2, to MAGSIAN using the definition suggested by [1] which will be briefly described here. This is done by projecting the agents onto MAGSIAN, then expand each individual game using the KBSC on to the MAGSIAN and after that compose all the games onto each other to form a complete game G^K . Lastly, all the observations are partitioned onto to each agent and serves as the reasoning about knowledge for each agent. This results in **Multi-agent Knowledge Based Subset Construction (MKBSC)**.

1) *Iterations*: The MKBSC can be used to iterate over a game G to generate **higher orders of knowledge**, which is knowledge about general knowledge [1]. At the 0:level (G^K) of knowledge, one reasons about ones own knowledge, at the 1:st level (G^{2k}) level of knowledge, one reasons about one's fellow agents 0:th level knowledge and ones own first level of knowledge, and can be more generally defined as n :th level of knowledge contains $n - 1$ levels of knowledge about the other agents and ones own n :th levels of knowledge and requires G^{K+n+1} expansions.

2) *MKBSC Strategy preservation*: This section will briefly discuss the strategy translation and preservation from expanded games using the MKBSC and KBCS back to the MAGSIAN respectively GIIN. This section has been explained more thoroughly in the papers of [1], [2] and [4].

This translation is based on the premise that the reachability and safety objective are translated into the expanded game.

Definition II.5 (Strategy preservation and translation). Let G be a MAGSIAN and G^K be the MKBSC expansion of that MAGSIAN. \mathcal{R} is the reachability objective for the MAGSIAN and let \mathcal{R}^K be its translated reachability objective for the expanded game. Then the following is true:

- 1) If there exists a winning strategy profile in G^K for \mathcal{R}^K , then there exist an equivalent winning strategy in G for the reachability objective \mathcal{R} [4].
- 2) This is true if the expanded game fulfills the PDK condition, where the PDK condition states that no two states (locations) in the MAGSIAN are indistinguishable for all agents [1].

H. Transducer

A transducer or Moore machine is a way to model a finite-memory strategy. The transducer receives an input which determines the next state of the transducer. Each input corresponds to a memory location and prescribed action for that specific state. It is formally defined as:

Definition II.6 (Transducer). Let a strategy α_i be represented by a Moore machine (transducer) defined as:

$$A_i = \{M_i, m_{0,i}, \mathcal{O}_i, \Sigma_i, \delta_i(m, \alpha_i, \mathcal{O}_i), \alpha_i\}$$

Where the parameters of the transducer are defined according to [6] as:

- i M_i is a finite set of **memory states**.
- ii m_0 is the **initial memory state** where $m_0 \in M_i$.
- iii \mathcal{O}_i are **observations** as defined previously.
- iv Σ_i are **action profiles** as defined previously.
- v δ_i is the **update function** defined as $\delta_i : M_i \times \mathcal{O}_i \rightarrow M_i$.
- vi α_i is the **memory-less strategy** as defined in II-E1 and is defined as following for a transducer $\alpha_i : M_i \rightarrow \Sigma_i$.

Where a memory state corresponds to a specified action to be performed after a specific observation [1] and $m_i \in M$.

The update function δ is a function that updates the current memory state in the transducer similarly to a transition. The update function operate based on the previous state, what the strategy prescribes for that specified observation and the observations made and is described in more detail in [1].

III. PROBLEM AND METHOD

A. Problem formulation

In this thesis we investigate the following:
Can a set of agents with no strategies, $Agent_a$, find winning strategies for themselves in a multi-agent game with imperfect information while knowing the strategy of a separate set of agents, $Agent_b$, if both sets of agents, $Agent_a$ and $Agent_b$, share winning objectives. Below, we formalize and define a type of game that fit the description above. Then a function is defined to transform the newly formalized game, MAGSIAN, to a MAGSIAN in order to be able to use the MKBSC as described in [1]. If a winning strategy can be found, it will be winning in the original game.

B. MAGSIAN

A multi-agent game with imperfect information against nature defined in II.3 with two categories of agents, one category with PDS and the other category with NPDS. This can be considered an expansion of the game type MAGSIAN.

Definition III.1 (Multi-agent game with given strategies with imperfect information against nature (MAGSIIAN)). Consider the MAGIIAN previously defined in II.3. Now the category of agents is partitioned into two sets, where a partition is equipped with transducer (strategy). The other partition of agents have no transducers (no strategy). This game can be viewed as a game graph tuple $G = (Agents, L, l_0, \Sigma, \Delta, O, \alpha_b)$ where:

- i $Agents = \{Agt_1, Agt_2, \dots, Agt_n\}$ are the **agents** playing vs Nature, these agents are partitioned into two set:
 - a) **Agents** of type **a**: $Agents_a = \{Agt_{a1}, Agt_{a2}, Agt_{a3}, \dots, Agt_{an}\} \subset Agents$ - These agents do not have a pre-defined strategy (α_a) (NPDS) and have **imperfect information**.
 - b) **Agents** of type **b**: $Agents_b = \{Agt_{b1}, Agt_{b2}, Agt_{b3}, \dots, Agt_{bn}\} \subset Agents$ Each agent has a pre-defined strategies (PDS). The strategies for $Agents_b$ are denoted α_b and can be modelled as the transducers $A_{b1}, A_{b2}, \dots, A_{bn}$.
- ii L is a finite set of **locations** describing the configurations of the game
- iii $l_0 \in L$ is the **initial location**
- iv $\Sigma = \Sigma_1 \times \dots \times \Sigma_n$ are the **action profiles** of the team.

The **action profiles** of group **a** are defined as:

$$\Sigma_{1a} \times \dots \times \Sigma_{na} = \Sigma_a \subseteq \Sigma$$

While the **action profiles** of group **b** are defined as:

$$\Sigma_{1b} \times \dots \times \Sigma_{nb} = \Sigma_b \subseteq \Sigma$$

Where Agents of type **a** and **b** have the following relationship:

$$\Sigma_a \cap \Sigma_b \neq \{\emptyset\}$$

- v $\Delta \subset L \times \Sigma \times L$ are the **transitions**, formulating all the edges between all locations in the game.
- vi $\mathcal{O} = \mathcal{O}_1 \times \dots \times \mathcal{O}_n$ are the **observation profiles** of the team of agents, where the observation profiles of the $Agents_a$ are: $\mathcal{O}_a = \mathcal{O}_{a1} \times \mathcal{O}_{a2} \times \dots \times \mathcal{O}_{an}$

C. Abstraction function

The abstraction function translates the original game to a new game where the agents with PDS:s are abstracted into nature, making the new abstracted game a Multi-agent game with imperfect information vs nature, also called MAGIIAN II.3. With this new MAGIIAN, we are able to look for strategies through higher levels of knowledge using the MKBSC II-G.

Definition III.2 (Abstraction function). Let a MAGSIIAN be a game graph tuple $G = (Agents, L, l_0, \Sigma, \Delta, \mathcal{O})$ and let the transducer A_b belong to the agents of category PDS (as defined in definition III.1) and represent their strategies (as defined in definition II.6). Then there exists an abstraction function ξ where

$$\xi\{G, A\} = \xi\{(Agents, L, l_0, \Sigma, \Delta, \mathcal{O}), A_b\} = G_{Abs}$$

where $G_{Abs} = (Agents_a, L_a, l_{a0}, \Delta_a, \mathcal{O}_a)$ is called the **abstracted game** and is of type MAGIIAN (II.3).

- i $Agents_a$ - are defined as $Agents_a$ in definition III.1
- ii L_a - The **locations** of the abstracted game are defined as:

$$L_a = L \times M_{b1} \times M_{b2} \times M_{b3} \times \dots \times M_{bn}$$

- iii l_{a0} - are the **initial locations** of $Agent_a$ in the abstracted game.

$$l_{a0} = (l_0, m_{b10}, m_{b20}, m_{b30}, \dots, m_{bn0})$$

- iv Δ_a - Transfer function:

$$\Delta_a \subseteq L_a \times \Sigma_a \times L_a$$

- v \mathcal{O}_a - are the **observations** of $Agents_a$ defined in III.1.

1) *Pruning*: Since every location $l_i \in L$, corresponds to a set of tuple memory locations $m_i \subseteq M_b$ in the transducer according to the rule that each location has at the most **Actions** ^{$|Agents_b|$} number of memory states associated with it, the following pruning can be done:

If a location l_x , has no associated memory state m_y to that location, the set $\{l_x m_y\}$ is considered unreachable and pruned away. When all the locations $\forall l_x \in L$, have gone through this pruning action, the resulting $L_a = L \times M_b$ consists only of pairs of locations and memory states that have a reachable location associated with each prescribed action.

$$L_a = \{l_x \in L | m_y \in M_b, (l_x, \sigma, l'_x) \in \Delta, l_x \in m_y, l'_x \in m'_y\}$$

IV. RESULTS

A. Game example

To further illustrate the core concepts of the thesis, we will explore an example demonstrating the interaction between MAGSIIAN, abstraction function and strategy synthesis for single player games with imperfect information explored in [4].

Two agents Agt_a and Agt_b work together to sort boxes based on if a loaded elevator contains one or two boxes. However, the two agents are unaware of precisely how many boxes are inside the elevator, which is random. Agt_a is in charge of raising or holding the elevator $\{raising, holding\} = \{R, H\}$. Whereas Agt_b is in charge of sorting the boxes to the left if the elevator contains two boxes, and to the right if it contains one box $\{left - push, right - push\} = \{LP, RP\}$. However, Agt_b is not too good at its job and has decided to switch action every turn. To start the elevator, both agents have to turn on the machinery, which is done using actions $\{start, start\} = \{S, S\}$.

The engine powering the elevator generates a constant amount of force which is enough to push the elevator a certain length. It begins by pushing one unit of length with the actions $\{S, S\}$. If there are two boxes in the elevator, it stops midway and have to be activated again to reach just below the hatch. If there is one box, the elevator is pushed just below the hatch without stopping midway. Either way, the Agt_a does not know how many boxes there are in the

elevator at the start, it can only observe if the elevator is below the hatch or not, and if the elevator is in its starting position or not. Ag_t_b on the other hand, knows the exact location of the elevator, because he is the son of Zeus and has X-ray vision. However, he does not explicitly know the amount of boxes in the elevator. This situation can be seen below in fig.3 and fig. 4 and be modeled as a game, seen in fig. 5.

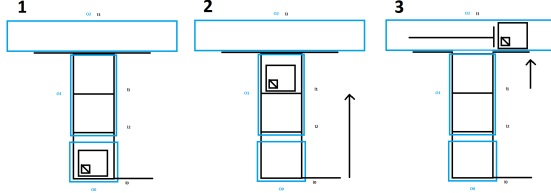


Fig. 3. Example, when one box is loaded on the elevator. These three figures explain the course of the game where: $1 = \{l_0, m_0\}$, $2 = \{l_1, m_3\}$, $3 = \{l_3, m_5\}$ in fig. 5

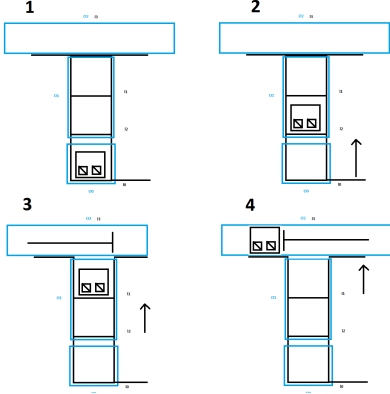


Fig. 4. Example when two boxes are loaded into the elevator, where: $1 = \{l_0, m_0\}$, $2 = \{l_2, m_4\}$, $3 = \{l_1, m_3\}$, $4 = \{l_3, m_5\}$ in fig. 5

The game declared above can be viewed as a MAGSIAN consisting of two categories of agents, Ag_t_a and Ag_t_b where agents of category b (PDS) has a finite-memory strategy (α_b) in the form of a transducer (A_b). In addition, Ag_t_b have perfect information regarding the game. The goal of the game is to reach the location $\{l_3\}$, which represents that the boxes are sorted correctly. The start of the game is when both Ag_t_s pick the actions $\{S, S\}$. If they end up in state $\{l_1\}$, below the hatch, or $\{l_2\}$, midway through the hatch, is determined by nature which is non-deterministic. The locations $\{l_1, l_2\}$ belong to the same observation (\mathcal{O}_1) for Ag_t_a . The given strategy for Ag_t_b is, after start, alternating between picking LP and RP , starting with LP after action S . This strategy can be seen in the transducer A_b in fig. 6. This game can be viewed as a graph, which is depicted in fig. 5. In fig. 5 the red dotted lines between states indicates that they belong to the same observation, in this case $\{\mathcal{O}_1\}$. The agent with NPDS are able to distinguish the other locations, $\{\mathcal{O}_0\} = \{l_0\}$ and $\{\mathcal{O}_2\} = \{l_3\}$.

In a generic MAGSIAN there are multiple Ag_t_a and Ag_t_b in Ag_t_a respectively Ag_t_b . When there are more than a

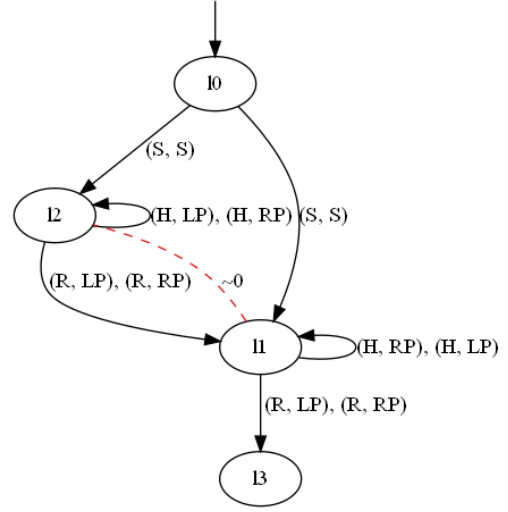


Fig. 5. MAGSIAN representation, where the red dotted line represents the observation \mathcal{O}_1 and Ag_t_a 's action is illustrated as the first action in the action profile (S,S). Conversely, Ag_t_b 's actions is the second action in the action profile pair.

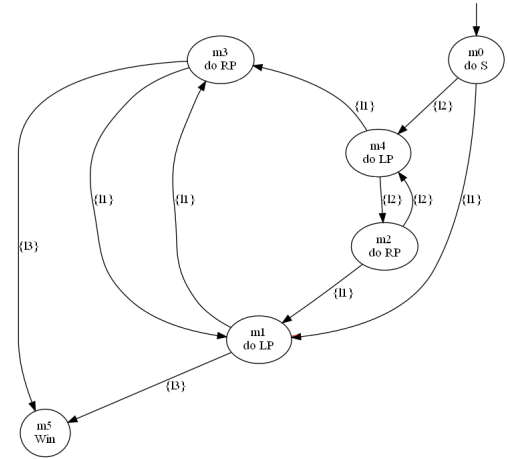


Fig. 6. Transducer for Ag_t_b in the MAGSIAN, as seen in III.1 the agent has perfect information.

single Ag_t_a in Ag_t_a one must use MKBSC due to there being multiple agents in the translated MAGSIAN. In this particular example, due to Ag_t_a consisting of only one agent it is more appropriate to use the KBSC, since, as previously stated, the MKBSC is a generalization of the KBSC. The MKBSC is not used since the abstracted game is a single player game. Since the strategy for Ag_t_b is known to Ag_t_a , we seek to abstract away Ag_t_b into nature and turn the Multi-player game into a single-player one, and search for a winning finite-memory strategy for Ag_t_a . This is accomplished by using the abstraction function defined in definition III.2, taking the abstracted game and then apply the KBSC construction on the GIAN and search for a memory-less winning strategy for Ag_t_a in the expanded game, resulting in a finite-memory strategy in the MAGSIAN.

B. MAGSIIAN representation

The example previously mentioned can be represented as a MAGSIIAN game $G = (\text{Agents}, L, l_0, \Sigma, \Delta, \mathcal{O}, \alpha_b)$ where:

$$\begin{aligned} \text{Agents} &= \{\{Agt_a\}, \{Agt_b\}\} \\ L &= \{l_0, l_1, l_2, l_3\} \\ l_o &= \{l_0\} \\ \mathcal{O} &= \{\mathcal{O}_0, \mathcal{O}_1, \mathcal{O}_2\} \\ \Sigma &= \{S, LP, RP, R, H\} \\ R &= \{l_3\} \end{aligned}$$

C. Abstracted game

After applying the abstraction function on the MAGSIIAN Agt_b has been abstracted into nature, making the game a single player game of imperfect information vs nature and is shown in fig. 7. Below are the properties of the abstracted game.

$$\begin{aligned} \text{Agents} &= Agt_a \\ L &= \{\{l_0, m_0\}, \{l_1, m_1\}, \{l_1, m_3\}, \\ &\quad \{l_2, m_2\}, \{l_2, m_4\}, \{l_3, m_5\}\} \\ l_o &= \{l_0, m_0\} \\ \mathcal{O} &= \{\mathcal{O}_0, \mathcal{O}_1, \mathcal{O}_2\} \\ \Sigma &= \{S, R, H\} \\ \mathcal{R} &= \{l_3, m_5\} \end{aligned}$$

This transition has made the game a GIIAN, and this enables Agt_a to look for the existence of a memoryless strategy in an expanded game.

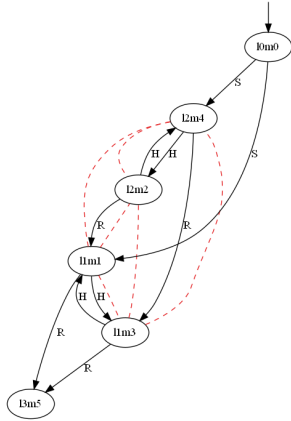


Fig. 7. Abstracted game G_{Abs} , Agt_b is abstracted into nature, and \mathcal{O}_1 is clearly visible and indicated by the red dotted line.

D. Expanded game

As previously mentioned, we use the KBSC expansion on the abstracted game G_{Abs} . This nets us the expanded game G_{Abs}^K which is depicted in fig. 8. In the expanded game G_{Abs}^K if a memory-less winning strategy is found, it can be translated to a winning finite-memory strategy in the abstracted game. Hence, our goal is to find a winning memory-less strategy for Agt_a and translate it back to the abstracted game G_{Abs} , and ultimately find a winning strategy in the MAGSIIAN G for

all *Agents*. If a winning strategy can not be found, then there can still exist a winning strategy, but this can not be found using the above-mentioned constructions [31].

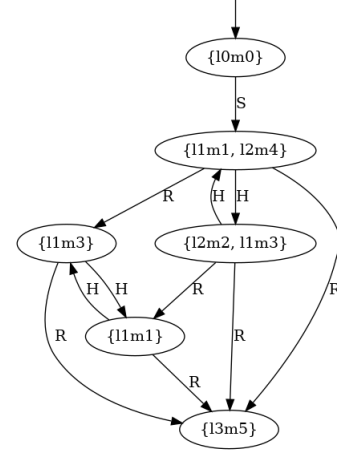


Fig. 8. KBSC expansion of G_{Abs} , the expanded game has perfect information.

E. Strategy translation

If a winning memory-less strategy has been found in the expanded game G_{Abs}^K , it will be returned to the abstracted game G_{Abs} according to definition II.5. One of the possible strategies found in the expanded game is seen in the fig. 9.

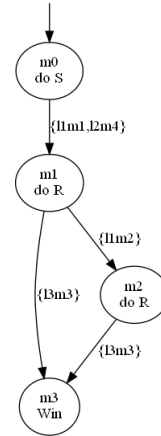


Fig. 9. Returned strategy for Agt_a , in the form of a transducer in the expanded game.

This specific transducer is then translated back to the abstracted game, G_{Abs} , resulting in the transducer seen in fig. 10. Since the transducer is winning in both the expanded and abstracted game, we need only translate the abstracted game's transducer to the MAGSIIAN. The transducer in fig. 9 and 10 differ only in locations and observations, since the abstracted and MAGSIIAN Agt_a are only able to use observations of the current and previous state of the game and cannot differentiate between all states they are able to use the same transducer. This notion hold true if each location $\{l_x, m_y\} \in L_a$, in the abstracted game, G_{Abs} can be collapsed to a location $\{l_x\} \in L$ in the MAGSIIAN and if each observation $o_i \in \mathcal{O}$ in G_{Abs} contains the same

information as in the MAGSIIAN. To illustrate these last two points, we can study the game example with this in mind:

If we start in location $\{l_2\}$ in the MAGSIIAN example and observe the action, (H, LP) , the next location observed will be $\{l_2\}$ again, this is observed by Agt_a as $\{\mathcal{O}_1\}$. In the abstracted game, we note the location $\{l_2, m_4\}$ that says we are in location $\{l_2\}$ in the MAGSIIAN where Agt_b will perform the action given by the transducer of memory location m_4 , if Agt_a then performs the action 'H' as before, we will now observe the location $\{l_2, m_2\}$, which following the same logic can be viewed as location $\{l_2\}$ again in the MAGSIIAN, and will be observed as $\{\mathcal{O}_1\}$ again. Since no new information is created using the Abstraction function and the observations are the same, the transducer used in G_{Abs} will be able to be equivalent to the transducer in G , thus we can say that,

$$A_{Abs,a} = A_a$$

The logic performed above can be viewed as collapsing $\{l_2, m_4\}$ on the location $\{l_2\}$, making the observations in the different game equivalent.

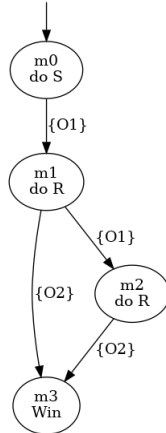


Fig. 10. Translated transducer from expanded to abstracted game and ultimately also the transducer for the MAGSIIAN G

The resulting game for the abstracted MAGSIIAN will now play the game according to the strategy the two transducers prescribes for each agent, an example can be seen in table I.

Table I
FULL PLAY OF GAME EXAMPLE

Location	Agt_a	Agt_b	Obs
l_0	S	S	\mathcal{O}_0
l_2	R	LP	\mathcal{O}_1
l_1	R	Rp	\mathcal{O}_1
l_3	win	win	\mathcal{O}_2
Location	Agt_a	Agt_b	Obs
l_0	S	S	\mathcal{O}_0
l_1	R	LP	\mathcal{O}_1
l_3	win	win	\mathcal{O}_2
-	-	-	-

V. DISCUSSION

A. Future work

1) *Proving the content:* Since this thesis makes claims about the concepts presented, there is a need for rigorous

mathematical proofs of the content. The authors believe that the method presented in here could prove useful in solving the general case of two player games versus nature that today are unsolvable. This due to the fact that the first agents' strategy can be fixed and one can then try to find a strategy for the second agent that reaches the common objective. This allows for an iterative approach to solving the problem instead of a deductive approach.

2) *Formalize strategy translation from MAGIIAN to MAGSIIAN:* Due to the timeframe of the thesis course, the authors were unable to formally define the strategy translation from MAGIIAN to MAGSIIAN, if this translation can be mathematically formalized, this would help to further validate the formulized postulates of the thesis.

B. Search for winning memory-less strategies

Because the study investigates finite-memory strategies and its translations, it can be explored if the concepts of the thesis holds true for memory-less strategies in G as well. This is useful due to memory-less strategies requiring less resources than finite and perfect-recall strategies, and the fact that it impose stricter condition for proofs, ergo if a proof holds for memory-less strategies it will also hold for finite-memory and perfect-recall strategies.

VI. CONCLUSION

A new game structure, MAGSIIAN, was proposed that accommodates agents with PDS and agents with NPDS. Furthermore, an abstraction function to transform a MAGSIIAN to an abstracted game, a MAGIIAN, was proposed and defined. This abstractions function allows the MAGIIAN to explore if a finite-memory winning strategy could be found using MKBSC. If a memory-less winning strategy can be found for the expanded game, then a finite-memory strategy can be used in the abstracted game. Additionally, a way to reverse transform the finite-memory strategy to the MAGSIIAN from the abstracted game is proposed by collapsing the memory-states in the abstracted game onto the original locations in the MAGSIIAN. This strategy translation proposed in the example has two conditions to fulfill in order for the strategy to be transferable from the abstracted game to a MAGSIIAN. This is all illustrated in a basic example of a two player game with strategy with imperfect information against nature, in which the concepts proposed are utilized and demonstrated. This is not proven, but rather claims made based on intuition and knowledge of the field.

ACKNOWLEDGMENT

The authors would like to thank our appointed supervisor Dilian Gurov whom without this thesis would not be possible, who guided and helped us understand the concepts of his paper [1], which gave us a fundamental understanding of key aspects within the field which lead us to the area within which the paper is written in. We would also like to thank Jakobsson and Nylén for the construction they built, which was used to create the graphs used throughout the rapport and helped illustrate key aspects and examples [7].

REFERENCES

- [1] D. Gurov, V. Goranko, and E. Lundberg, “Knowledge-based strategies for multi-agent teams playing against nature,” *Artificial Intelligence*, vol. 309, p. 103728, May, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0004370222000686>
- [2] E. Lundberg, “Collaboration in multi-agent games: Synthesis of finite-state strategies in games of imperfect information,” Msc. thesis, KTH, Stockholm, Sweden, 2017.
- [3] L. Doyen and J.-F. Raskin, “Games with imperfect information: theory and algorithms,” *Lectures in Game Theory for Computer Scientists*, vol. 10, 2011.
- [4] J.-F. Raskin, T. A. Henzinger, L. Doyen, and K. Chatterjee, “Algorithms for omega-regular games with imperfect information,” *Logical Methods in Computer Science*, vol. 3, Nov. 2007.
- [5] H. Nylén and A. Jacobsson, “Investigation of a knowledge-based subset construction for multi-player games of imperfect information,” Bsc. thesis, KTH, Stockholm, Sweden, 2018.
- [6] D. Kozen, *Automata and Computability*, ser. Undergraduate Texts in Computer Science. Springer New York, 2007. [Online]. Available: https://books.google.se/books?id=8IKyxS8_CN0C
- [7] H. Nylén and A. Jacobsson, “mkbsc.” [Online]. Available: <https://github.com/helmernylen/mkbsc>

Playing the Fox Game With Tree Search: MCTS vs. Alpha-Beta

David Ye and Jacob Trossing

Abstract—The forefront of game playing Artificial Intelligence (AI) has for the better part of 21st century been using an algorithm called Alpha-Beta Pruning (Alpha-Beta). In 2017, DeepMind launched a new AI, based on the algorithm Monte Carlo Tree Search (MCTS), which defeated the former Alpha-Beta based chess AI champion Stockfish. This accomplishment fueled up more excitement and interest for using MCTS to develop more complex and better performing game playing AI.

This paper aims to compare the strengths of MCTS and Alpha-Beta by allowing them to play against each other in a classic game with no available robust AI - the Fox Game.

The results showed an evident victory for the Alpha-Beta AI. Therefore, Alpha-Beta is the better suited algorithm for developing a simple AI for the Fox Game. Further optimizations would enhance the performance of both algorithms but it is unclear which of the algorithms would benefit from it the most.

Sammanfattning—Framkanten av Artificiell Intelligens (AI) som spelar spel har i större delen av 2000-talet använt sig av en algorithm vid namn Alpha-Beta-beskränning (Alpha-Beta). Denna bedrift höjde intresset för att använda MCTS i syfte att utveckla mer komplexa och bättre spelande AI.

Denna rapport har som mål att jämföra styrkor hos MCTS och Alpha-Beta genom att låta dem spela mot varandra i ett klassiskt spel utan någon tillgänglig AI - Rävspelet.

Resultaten visade på en klar seger för Alpha-Beta AI:n. Därför är Alpha-Beta den bättre lämpade algoritmen för att skapa en simpel AI. Fler optimeringar hade förbättrat spelstyrkan hos bägge algoritmerna med det är oklart vilken av algoritmerna som hade gynnat mest utav det.

Index Terms—Artificial Intelligence, Monte Carlo Tree Search, the Fox Game, Alpha-Beta Pruning, Asymmetrical Game, Perfect Information Game

Supervisors: Mika Cohen and Farzad Kamrani

TRITA number: TRITA-EECS-EX-2022:181

I. INTRODUCTION

A common method to learn new strategies of a game is to develop an artificial intelligence to play. Famous examples of games with robust AI include chess and Go. Both of these games are classified as perfect information games, which is a category of games where the game only depends on the players' choice of moves. All information from previous events are available and all possible futures can be deduced [1].

Alpha-Beta Pruning (Alpha-Beta) is a tree searching algorithm, which has historically been the standard algorithm used in AI playing perfect information games. Traditional chess engines ranging from IBM's Deep Blue, which in 1997 became the first computer to beat a chess world champion, to the 2016 chess engine world champion Stockfish utilize Alpha-Beta.

In late 2017, a newly created AI managed to defeat Stockfish in the 2017 Chess Engine Championship. This new engine – known as AlphaZero relies on a deep neural network to guide another search algorithm called Monte Carlo Tree Search (MCTS) to find the most promising moves [2]. This new algorithm proved to be a promising alternative to the traditional Alpha-Beta algorithm with AlphaZero, managing to master two other notable perfect information games – Shogi and Go. This has given DeepMind, the creator of AlphaZero, hope to create general-purpose learning systems to play any perfect information game. This learning system could then be expanded further to help solve important and difficult real-world problems [2].

In order to acquire more insights about these two algorithms, this research will be conducted on a perfect information game which still lacks a robust AI – the Fox Game.

The Fox Game is a classic Scandinavian strategy game which according to [3] has existed for more than 700 years. It is played by two players, one playing as 20 sheep with the goal to fill a 3x3 square with sheep, and the other playing as 2 foxes with the goal of removing the sheep from the board to prevent them from reaching their goal. It is heavily asymmetrical in starting position, goals and rules. This feature does not exist in many other classical games often studied on such as chess, go and shogi. This, combined with its cultural relevance, strategic depth and lack of strong AI makes it an interesting game to research.

Following the introduction, an overview of how to play the Fox Game is presented. The theoretical background follows and explains the algorithms MCTS and Alpha-Beta. Subsequently, the method chapter discusses the details concerning the implementation of the algorithms and optimizations as well as how the experiments were conducted. Then, the results of the experiments are presented and discussed. In addition, ideas for future research are also discussed. Lastly, a conclusion will be drawn based on the discussions.

A. Problem statement

In this report, the performance of two prominent tree search algorithms, Alpha-Beta and MCTS, are compared in the classic strategy game the Fox Game. The goal of this comparison is to assess whether MCTS or Alpha-Beta is best suited for developing an AI to play the Fox Game.

II. THE FOX GAME

Fox Games are a class of board games for two players, where one plays the fox and the other plays the geese/sheep.

The objectives vary between different versions. This paper will study the Scandinavian version seen in figure 1. The following text presents all information the reader needs to understand how this version is played followed by a play example.



Fig. 1. The board of the traditional Scandinavian version of the Fox Game.

A. Brief description of the game

As in many other two player games, the players take turns making their moves, starting with the foxes. The game board can be seen in figure 1. The starting board consists of 2 foxes and 20 sheep. Each piece is given a pre-determined **starting location** seen in figure 2: F denotes a fox, S for sheep, O for empty space and \emptyset for empty diagonal space. An empty space is an unoccupied space and a diagonal space is a space where the foxes can move diagonally. The sheep can only move up, left and right. However, the foxes can move up, down, left, right and the four additional diagonal directions if they occupy a diagonal space. Additionally, foxes can jump over a sheep if the board space behind is empty, resulting in the sheep being removed from play. Whenever a fox is able to jump, the fox has to jump, **continuously** until it is not able to. Also, a fox F is removed from play if there are no available move for the fox F. Only one piece can be moved each turn and a piece

$$\begin{bmatrix} & & F & O & F & & \\ & & O & \emptyset & O & & \\ \emptyset & O & \emptyset & O & \emptyset & O & \emptyset \\ S & S & S & S & S & S & S \\ S & S & S & S & S & S & S \\ & & S & S & S & & \\ & & S & S & S & & \end{bmatrix}$$

Fig. 2. Starting positions for the Fox Game.

can not move to a board space occupied by another piece. The **terminal conditions** are as follows:

- The sheep **win** if either all board spaces in the 3x3 field at the top are occupied by sheep or all foxes are removed from play.
- The foxes **win** if there are less than 9 sheep on the board.
- A game will conclude as a **draw** only if the number of turns exceed 200 or if both players repeat a two move sequence twice.

B. A basic example

The example below shows how a set of turns can be played. Assume the given board state on an artificial board. The board spaces (x,y) range from $0 \leq x \leq 2$ and $0 \leq y \leq 2$ with origin placed at top-left corner.

- A fox at (0, 0)
- A sheep at (2, 1)
- A sheep at (1, 2)

Turn 0 - The assigned locations

$$\begin{bmatrix} F & O & \emptyset \\ O & \emptyset & S \\ \emptyset & S & \emptyset \end{bmatrix}$$

Turn 1 - Sheep turn: Sheep moves (1,2) to (1,1)

$$\begin{bmatrix} F & O & \emptyset \\ O & S & S \\ \emptyset & \uparrow & \emptyset \end{bmatrix}$$

Turn 2 - Fox turn: Fox jumps two times from (0, 0) to (2, 2) to (2, 0)

$$\begin{bmatrix} \swarrow & O & \emptyset \\ O & \searrow & S \\ \emptyset & O & F \end{bmatrix}$$

$$\begin{bmatrix} \emptyset & O & F \\ O & \emptyset & \uparrow \\ \emptyset & O & \uparrow \end{bmatrix}$$

III. THEORY

The theory required to understand the results and discussion will be presented in this chapter, starting with a section briefly discussing why AI in games is an interesting research topic. It is followed up by theoretical explanations of the tree search algorithms MCTS as well as Alpha-Beta Pruning.

A. Artificial intelligence in game playing

Artificial intelligence (AI) or Computational Intelligence (CI) is according to [4] "the study of the design of intelligent agents", with intelligent referring to the agents acting in a way that help them complete their goals. A well defined goal often associated with intelligence in humans is winning at board games such as chess. This makes it interesting to see if we can design intelligent agents that are able to win. As a result, a large amount of research has been conducted around the topic. Today, many AI-programs have been developed which have defeated world champions at various board games.

However, there is still much more to research in this topic. While there are few games where AI has yet to beat human players, the pursuit for intelligent agents has not ceased. Large AI game playing tournaments are now held to find the most intelligent agents. There are also many strategically interesting games such as the Fox Game where currently no strong and robust AI exists.

B. Monte Carlo Tree Search

Monte Carlo Tree Search or MCTS is a best-first search algorithm which combines tree search with Monte Carlo simulations [5]. The fundamental concept of this algorithm is to find the most promising move by simulating matches played out using random moves. The idea is that better moves should on average lead to more wins and more simulations should even out the randomness. These moves are added to a tree which expands as the algorithm continues. This results in not only the available moves being looked at, but the opponent's responses and our re-responses etc. are also being considered.

The method consists of four phases - **selection**, **node expansion**, **playout** and **backpropagation**.

The selection is the procedure to traverse from the root to a leaf node. Each node S_i represents a game state; the children of each parent are the possible subsequent game states of the parent state. The root represents the **current** game state for which the algorithm looks to find the best move. The nodes possess two variables which are: a value v_i of the state and the number of times the node has been visited n_i . The value v_i represents the total amount of won payouts node i has.

In order to identify the most promising move to explore further at a given game state, the function

$$UCB1(S_i) = \bar{V}_i + C \sqrt{\frac{\ln n_{p_i}}{n_i}}$$

discussed in [5] is introduced where $\bar{V}_i = \frac{v_i}{n_i}$, n_{p_i} stands for the number of visits of the parent node of S_i and C denotes the exploration parameter - usually chosen to be the theoretical value $\sqrt{2}$ described in [6]. The first term corresponds to exploitation and is significant whenever the state leads to a high average evaluation. The second term corresponds to exploration and is significant whenever the node has a low visit count, making up for the uncertainty from the Monte Carlo simulations. The traversal through the tree occurs by always picking the node which **maximizes** the value from UCB1 until a leaf node is reached. This simulates each player playing the currently most promising moves which reduces the exploration of weak responses.

The selection phase is followed by the expansion phase. During a node expansion, the new leaf node selected in the selection phase is expanded by adding all subsequent board states to it as children. The value v_i and visited count n_i of these new leaf nodes are set to 0. This phase is followed up by the playout phase.

During the playout, a simulation is run by self-play. Random or semi-random moves are generated for both players until a terminal state has been reached (i.e. end of game). A value r will then be returned, which represents the result of

the playout. The playout is followed up by the final phase, backpropagation.

The objective of backpropagation is to propagate the result from the simulation r back to all of the previous traversed nodes, including the root node. To account for players alternating their turns, the value r_i added to v_i of the i 'th visited node is

$$r_i = 1 - r_{i+1}.$$

The number of times visited, n_i , of all the traversed nodes are also incremented by one. After this final phase, a new iteration will occur, starting at the first phase. This continues as long as the allotted time is not exceeded.

When the turn timer has been exceeded, the move leading to the most promising board state is returned. In our implementation, this is the child of the root node with the most visits as this guarantees a move that has been evaluated and explored multiple times.

C. Alpha-Beta Pruning

Alpha-Beta Pruning is an optimization technique to reduce the search space of minimax, which is a tree-based algorithm. Minimax is used in sequential games to determine the most promising move at each game state with the assumption that the opponent plays optimally. Sequential games refer to games where the players alternate their turns. As the name of the algorithm suggests, there is a minimizer (the opponent) and a maximizer (the player). The children of max-nodes (maximizers) are min-nodes (minimizers) and vice versa. The depth for a node n_i is the number of edges between n_i and the root, which in this context **serves** as the number of moves from the current game state to the node. Every game state has a value that is obtained from an evaluation function. An evaluation function receives a board state as input and returns a value as output. Thus, given a depth value d , the AI will pick a move that leads to the highest board state value in d turns assuming optimal responses.

Alpha-Beta Pruning introduces two new parameters - alpha and beta. Alpha is the best value that the maximizer has found for that depth and above. Beta is the best value that the minimizer has found for that depth and above. The core idea is to, with the use of the newly introduced parameters, cut off branches in the tree to reduce search of the space as better values have already been found.

Furthermore, an iterative deepening has been implemented to allow limit the turn timer of Alpha-Beta Pruning. Without it, the algorithm searches until a given depth, taking an inconsistent amount of time to decide a move. The process is as follows:

- 1) Begin at depth = 0 and calculate the best move
- 2) Increment the depth and calculate the best move
- 3) Iterate until the thinking time runs out and return the best move found for largest completed depth.

D. Difference in tree structure between MCTS and ALpha-Beta

As mentioned earlier, MCTS finds the best move from a given game state by simulating many games from that state to

the end. The exploration of the tree is biased towards nodes which have few visits and nodes that yield a high value from UCB1. Therefore, the resulting tree is asymmetric and the depth for promising nodes are deeper than other nodes. In contrast to MCTS, Alpha-Beta explores every possible move and game state to the given depth, resulting in an even tree where all leaf nodes have equal depth.

IV. METHOD

This chapter introduces the implementation of the project, pseudocodes for the algorithms and three optimization techniques used in the project - namely evaluation, cutoff and bitboards. Finally the different experiments conducted are described.

A. Implementation

The game is implemented, along with the discussed algorithms MCTS and Alpha-Beta Pruning in **Python 3.9**. The code can be found here: <https://gits-15.sys.kth.se/jacobtr/mcts-fox-game>.

The pseudocode for MCTS is presented in Algorithm 1. Here *timeLeft()* is a function which returns true if there is still time left on the turn, otherwise false. *selectChild(node)* returns the child node with the highest UCB1 score. *expandNode(node)* adds all the children of the node to the tree. The *playOut(node)* method performs the playout step, simulating the game using random moves and returns a result *r*. *BackPropagation(r, node)* updates the node with the given result *r*. *mostVisitedChild(node)* returns the child node which has been visited the most times.

Algorithm 1 MCTS

```

1: Initialize a tree  $T$ 
2: while timeLeft() do
3:    $currentNode \leftarrow root$ 
4:    $visitedNodes \leftarrow$  Empty list
5:   // Tree traverse and select child with UCB1
6:   while  $currentNode$  is not a leaf node do
7:      $visitedNodes.append(currentNode)$ 
8:      $currentNode \leftarrow selectChild(currentNode)$ 
9:   end while
10:   $visitedNodes.append(currentNode)$ 
11:  // Node expansion
12:   $expandNode(currentNode)$ 
13:  // Simulation
14:   $r \leftarrow playOut(currentNode)$ 
15:  // Backpropagation
16:   $visitedNodes = reverse(visitedNodes)$ 
17:  for  $node$  in  $visitedNodes$  do
18:     $Backpropagation(r, node)$ 
19:     $r \leftarrow 1 - r$ 
20:  end for
21: end while
22: return  $mostVisitedChild(root)$ 

```

The pseudocode for Alpha-Beta Pruning is presented in Algorithm 2. Here *d* is the depth in which the algorithm should

search to and *timeLeft()* returns true (T) if there is time left on the turn, otherwise false (F). While this method only returns the highest value found, the best move can be obtained from a function *ABMaxMove(node, depth, alpha, beta)*. The function would mimic the code describing *maximizer* and also keeps track of which move has resulted in the highest value. Afterwards, it returns the move instead of the value. With this function, we are able to call the algorithm through *ABMaxMove(root, 0, $-\infty$, ∞)*.

Algorithm 2 Alpha-Beta Pruning

```

1: procedure AB( $node, depth, maximizer, alpha, beta$ )
2:   if  $depth == d$  or  $node$  is terminal node then
3:     return  $Evaluation(node)$ 
4:   end if
5:   // Maximizer and minimizer are alternating turns
6:   if  $maximizer$  then
7:      $bestValue \leftarrow -\infty$ 
8:     for  $child = 1, 2, \dots, N$  do
9:        $value \leftarrow AB(child, depth+1, F, alpha, beta)$ 
10:       $bestValue \leftarrow \max(bestValue, value)$ 
11:      if  $bestValue \geq beta$  or  $not timeLeft()$  then
12:        return  $bestValue$ 
13:      end if
14:       $alpha \leftarrow \max(alpha, bestValue)$ 
15:    end for
16:   else
17:      $bestValue \leftarrow \infty$ 
18:     for  $child = 1, 2, \dots, N$  do
19:        $value \leftarrow AB(child, depth+1, T, alpha, beta)$ 
20:        $bestValue \leftarrow \min(bestValue, value)$ 
21:       if  $bestValue \leq alpha$  or  $not timeLeft()$  then
22:         return  $bestValue$ 
23:       end if
24:        $beta \leftarrow \min(beta, bestValue)$ 
25:     end for
26:   end if
27:   return  $bestValue$ 

```

B. Evaluation and rewarding methods

An evaluation function is a function which maps a board state to a value, indicating how favored a player is in a game state. The evaluation function e_s used in this game yields a value in the interval $(0, 1)$, which represents the estimated win rate for the sheep. Likewise, the estimated win rate for the foxes in any game state is obtained from subtracting the estimated win rate of the sheep from 1, $e_f = 1 - e_s$. The function is designed to evaluate from the following two principles:

- Evaluate the **positions** of the sheep by giving every board space an assigned value. The values are gradually increasing from the bottom to the field.
- Evaluate **number of pieces** left for each player.

The first principle arises from one of the win conditions - fill the field with sheep. Therefore, the sheep should always

0	0	250	250	250	0	0
0	0	190	190	190	0	0
120	120	120	120	120	120	120
30	30	30	30	30	30	30
20	20	20	20	20	20	20
0	0	14	14	14	0	0
0	0	13	13	13	0	0

Fig. 3. $E(x, y)$ for the entire board.

seek to move towards the field. In order to estimate a value based on the positions, the function

$$e_1 = \sum_{x=0}^6 \sum_{y=0}^6 \frac{E(x, y)s(x, y)}{C_e}$$

is introduced, where $s(x, y)$ yields a value of 1 if the board space (x,y) is occupied by a sheep and 0 otherwise. The constant C_e exists to keep the generated values small and was obtained through experiments described later. It was chosen to be $C_e = 2200$. $E(x, y)$ yields the assigned value on the board space (x,y) and is shown in figure 3.

However, this principle does not take the fox positions into consideration. Thus, a second principle is needed. A higher amount of sheep on the board will increase the evaluated value e_s . Conversely, more foxes will decrease it. This principle will help the function to assess fox positions indirectly. Fewer sheep is caused by fox jumps which is a good move by the foxes. Thus, this principle also assists the AI to predict bad sheep positions. The equation

$$e_2 = 0.72 \ln(0.1n_s) + 0.3(2 - n_f)$$

yields a value based on the second principle, where n_s and n_f are the number of sheep and foxes left on the board. The constants in the first term and the logarithm are chosen from experiments and observations, where e_2 is thought to be following a logarithmic curve as losing sheep matters more if fewer sheep are left on the board. The second term evaluates the number of foxes and the coefficient is a parameter determining the value of a fox. In addition, the values in the first term were selected such that e_2 returns a value 0.5 at the beginning of a game. A combination of these functions give rise to the final evaluation function

$$e_s = \frac{7}{9}e_1 + \frac{2}{9}e_2$$

with the weights chosen to be $\frac{7}{9}$ and $\frac{2}{9}$. To ensure the evaluation function returns a value existing in the interval (0,1), all evaluations smaller than 0 are set to 0.001 and larger than 1 set to 0.999. The experiments to determine the parameters in the evaluation function are conducted through a single elimination tournament system. These parameters include the coefficient value for the foxes, the weights, the evaluation board $E(x, y)$ and the constant C_e . In order to determine a value for a parameter p_i , all other parameters were fixed, and all AI participants had a different value of the parameter p_i . Each matchup between two AI consisted of 100

games with 50 matches on each side, and the highest combined win rate was determined to be the winner. The value of the winner AI of the tournament was later selected to be in the final function e_s .

C. Cutoff

A single game can end after many turns. During the payout phase in the MCTS algorithm these games are bottlenecks for predicting prominent moves and can reduce number of simulations significantly. Furthermore, randomness favor the foxes as it is easy to take advantage of a bad sheep move. By introducing cutoff, these lengthy games can be avoided. It would also diminish the effects of randomness. Evaluation cutoff assesses the game prematurely using an evaluation function and yields a value given by e_s . This strategy is known as *fixed-length cutoff*, which stops the playouts after a fixed amount of moves d_c have been played from the current game state. The value $d_c = 14$ was selected in the same way as the parameters discussed in chapter IV-B.

D. Bitboards

A bitboard representation is a unique approach to represent the board and to make various operations in board games. In essence, the technique uses binary numbers to represent the board where each bit represents a board space. A bit with the value of 1 indicates an occupied board space and a value of 0 indicates an open space. Each binary number holds all information of all pieces of a certain type in a game state. Therefore, all information of a game state is given by all of these binary numbers for that game state. An example of how the starting board would be represented with two binary numbers can be seen in figure 4.

The main benefit of this representation in comparison to representing the board as a matrix or list is the **efficiency in time complexity**, allowing more simulations per second. With the use of binary numbers, many possible moves for many pieces in any game state is determined simultaneously. Otherwise, the moves would be calculated by iterating over all pieces using a for-loop. The method uses bitwise operators and a method called *bitwise shifting*. In short, bitwise shifting shifts all bits in a binary number to either left or right, creating a new binary number which represents a new game state. In conjunction with logical operators, these sets of operators make it possible to implement this optimization.

E. Experiments

The data were sampled from multiple experiments with different turn timers and matchups.

- 1) The **first** experiment involved MCTS and Alpha-Beta Pruning playing both parties 250 matches each, for a total of 500 matches at different turn timers ranging from 0.1-5 seconds.
- 2) The **second** experiment had a similar approach as the first experiment, with the difference being in number of matches and turn timers. The algorithms played as both

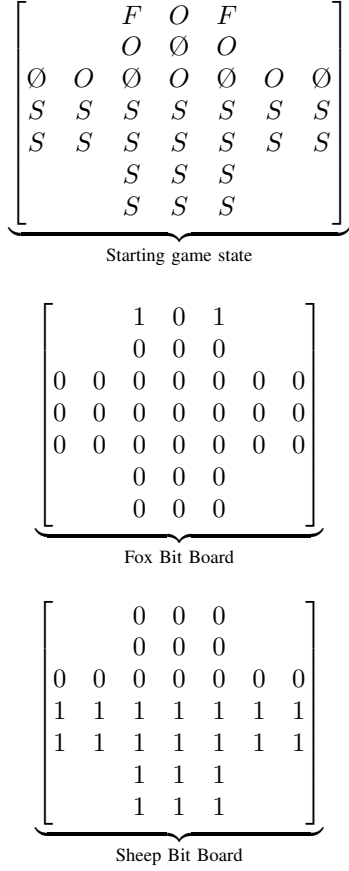


Fig. 4. Example of how the starting board can be represented using bitboards.

pieces 16 matches each, for a total of 32 matches with turn timer ranging from 10-60 seconds.

- 3) The **third** experiment examined the differences for 32 matches whenever the turn timer for MCTS was 10 times more than the turn timer for Alpha-Beta. The turn timer ranges for Alpha-Beta ranges from 1-6 seconds.
- 4) The **fourth** experiment examined MCTS' capabilities to play against humans. MCTS played a total of 5 games as both pieces for a total of 10 games. The games were played against 5 different opponents.

The purpose of the first two experiments was to analyze the performance of the two algorithms for different turn timers. The third experiment aims to emulate a parallelized MCTS to see how further optimizations would affect game playing strength. The fourth experiment examines MCTS' playing strength against humans and presents the strengths of the algorithms in a familiar context.

V. RESULTS

In this chapter the results from the experiments conducted will be presented starting with the results of MCTS playing versus Alpha-Beta Pruning followed by the results of MCTS playing versus human opponents. The black bars in the figures in this chapter represent the 95% confidence interval discussed later in chapter VI-B.

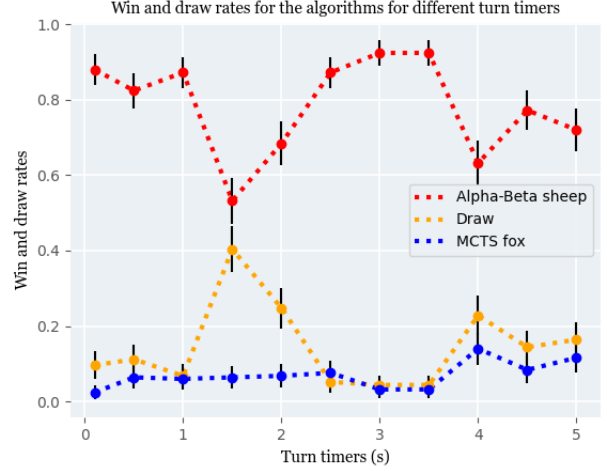


Fig. 5. Alpha-Beta as sheep and MCTS as fox win/draw rates. 250 matches were played.

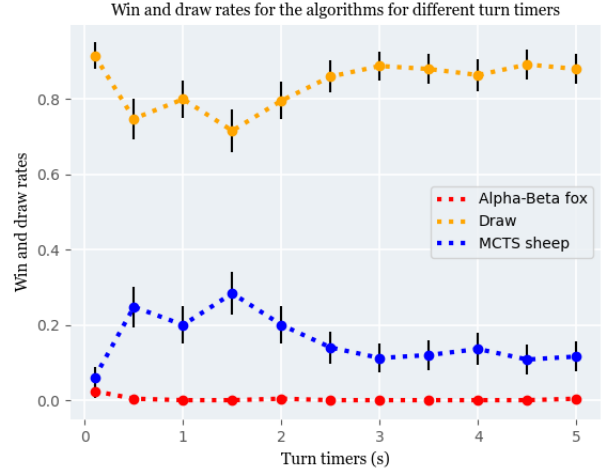


Fig. 6. Alpha-Beta as fox and MCTS as sheep win/draw rates. 250 matches were played.

A. MCTS vs. Alpha-Beta Pruning

The win rates for the MCTS and Alpha-Beta matchups are shown in six figures. In the first two figures 5 and 6, depicting the first experiment, both algorithms have a higher win rate as sheep than foxes at every turn timer. MCTS sheep can be observed to decline in win rate at a higher turn timer whereas Alpha-Beta sways back and forth at a win rate of around 80 percent. Alpha-Beta has a higher average win rate if both graphs are taken into consideration.

The second experiment is shown in figures 7 and 8. Alpha-Beta sheep wins more than MCTS sheep but MCTS fox wins more than Alpha-Beta fox.

Finally, the third experiment in figures 9 and 10 shows the win rate for Alpha-Beta whenever MCTS has ten times the turn timer Alpha-Beta has. The results show no significant difference for either side compared to the earlier experiments.

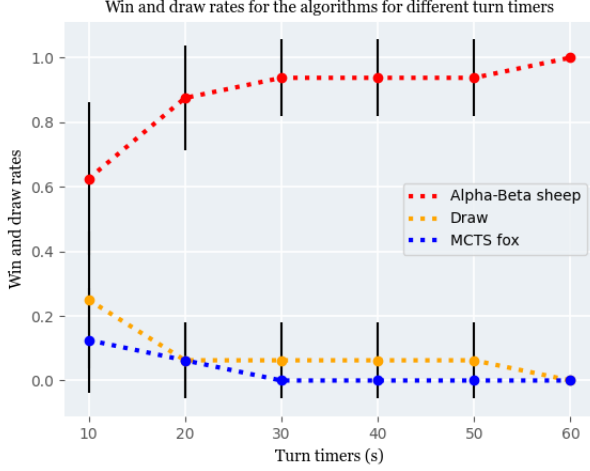


Fig. 7. Alpha-Beta as sheep and MCTS as fox win/draw rates. 16 matches where played.

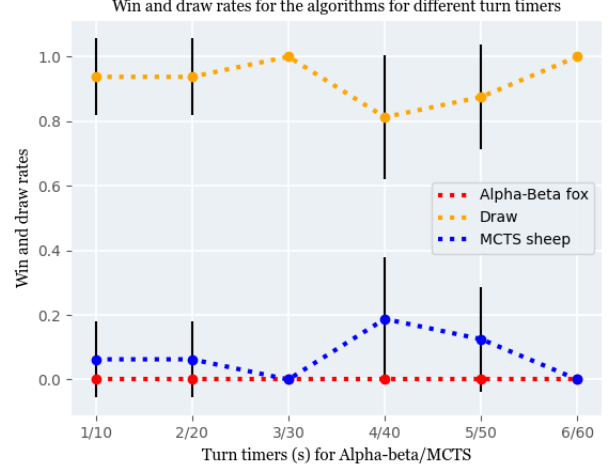


Fig. 10. Alpha-Beta as sheep and MCTS as fox win/draw rates where MCTS has 10 times more timer than Alpha-Beta. 16 matches where played.

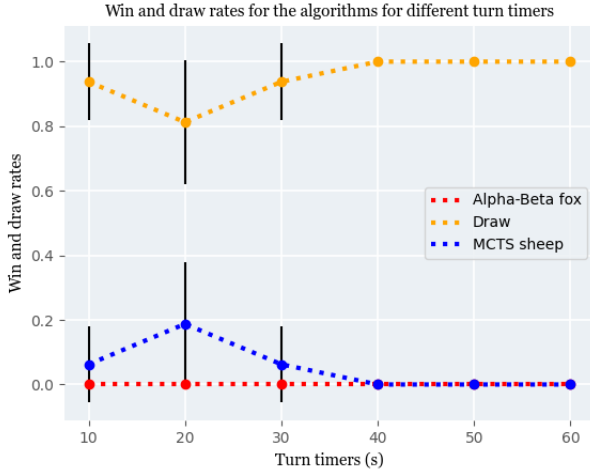


Fig. 8. Alpha-Beta as fox and MCTS as sheep win/draw rates. 16 matches where played.

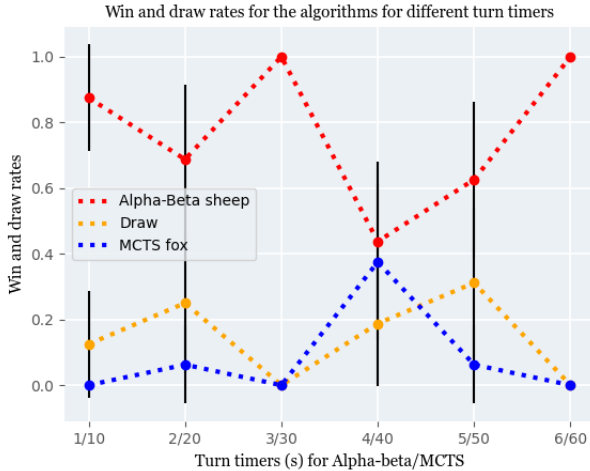


Fig. 9. Alpha-Beta as fox and MCTS as sheep win/draw rates where MCTS has 10 times more timer than Alpha-Beta. 16 matches where played.

B. MCTS vs. human opponents

The results from MCTS vs Human opponents can be seen in table I. The 10 games with 5 as each side, resulted in MCTS winning 6 games, losing 2 games and drawing 2 games.

VI. DISCUSSION

Finally, an analysis of the results will be conducted in this section. Statistical accuracy of the experiments will be presented as well as some observations regarding the fox game and ideas for further research.

A. Evaluation of results

As seen in figures 5, 6, 7 and 8, Alpha-Beta performed better than MCTS if the win rates of both pieces are taken into consideration. Alpha-Beta wins more as sheep although MCTS wins more as the foxes. However, the large discrepancy in the sheep performance of the algorithms indicates that it is most likely not because MCTS plays better as the foxes. Considering the fact where the playouts affect plays of both pieces, it should play equally well as both foxes and sheep. It is therefore reasonable to assume that MCTS' capability of forcing draws is remarkable. This can be noted from the results where Alpha-Beta does not manage to win a single game as foxes when MCTS receives more than 0.1 seconds to think.

The third experiment showed in figures 9 and 10 simulates a parallelized MCTS by giving MCTS 10 times the turn timer. The results surprisingly showed no significant improvement for MCTS. These results lead to the conclusion that Alpha-Beta is better than MCTS with the current implementations.

These results are surprising as Alpha-Beta has an exponential time complexity (searching for a move in a exponential growing tree) in contrast to MCTS which simulates games until the turn timer ends. Theoretically, a higher turn timer should therefore benefit MCTS more than Alpha-Beta. Hence, the results indicate improvements in only the performance are not enough for MCTS to close the gap on Alpha-Beta, unless

TABLE I
RESULTS FROM PLAYING MCTS VS. A HUMAN OPPONENT

MCTS Side	Games won	Games Lost	Games Drawn
Fox	3	2	0
Sheep	3	0	2

much larger turn timers or much more powerful hardware is used.

If the performance optimizations for MCTS do not yield other results, then it becomes interesting to analyze why Alpha-Beta outperforms MCTS. In consideration to the surprising results, it is important to remember that programming oversights can not be excluded. However, two factors were identified to affect the results.

The first factor involves the asymmetric properties of the game. As discussed in chapter IV-C, the randomness in the MCTS playout phase heavily favors the foxes. As a consequence, every simulated match is won by the foxes, which in turn leads to UCB1 identifying every move to be equally bad at every game state. The implementation of cutoff mitigated the randomness although it still affects the moves played during the simulations. Looking at Alpha-Beta, the asymmetry does not seem to affect it in a significant manner, which is one reason why Alpha-Beta played performed better than MCTS.

The second factor is an observation made from looking at the draw rates. MCTS sheep has more draws than Alpha-Beta sheep. During the simulations, it was observed in many matches where MCTS struggled to progress games, even in games in favor of the sheep. The MCTS AI could not take a short term loss and refused to sacrifice a sheep. These games always ended in draws. The asymmetry of the games enables this as the sheep are able to position themselves in a way such that the foxes cannot jump over them. In other words, the sheep are more proactive and the foxes are more reactive. However, it is unclear why only MCTS shows this behavior.

To put the playing strength of the algorithms in a familiar context, we had MCTS play against humans. The results, along with the observations seem to show that MCTS plays at a level of a strong human player, only losing two of the games played. Two of its biggest strengths are its methodical play and its level of patience. As the sheep, it leaves close to no openings for the foxes to jump over a sheep, although this patience also results in one of its biggest weaknesses. As just discussed whenever the sheep has amassed a big lead in a game, they are unable to progress the game forward, making moves which do not bring the sheep closer to their goal. This resulted in the MCTS sheep ending the games as draw in two favorable positions.

B. Statistical accuracy

Even though the game is deterministic in nature, the MCTS algorithm uses randomness in the playout phase. This leads to a randomness in how each games plays out and who wins the game. An error estimate has been calculated to understand how accurate the simulated win rates are. The error bars seen

in the figures in chapter V are all calculated using the formula

$$MOE = 1.96 \sqrt{\frac{wr_i(1 - wr_i)}{N}},$$

which calculates the margin of error corresponding to a 95% confidence interval as discussed in [7]. Here, wr_i is the win rate for the given match and N is the number of games played.

The error estimates are quiet large, especially for longer turn timers with only 16 matches as each piece.

Even with the large error estimates, the graphs show an evident result. None of the error bars overlap each other, which implies that Alpha-Beta still performs better even when considering the margin of error.

C. Observations about the Fox Game

Overall, the game heavily favors the sheep with remarkably few games ending with fox wins. One explanation for this one-sided result is the draw mechanism. If the sheep are at a losing game state, they can choose to position themselves next to walls to prevent foxes from jumping. The foxes cannot force the sheep to progress the game as long as they stay by the walls, which ends in a draw in most cases.

D. Future Work

There are many optimizations, which have not been implemented and would improve the performance of the algorithms. Recurring game states can be avoided with implementation of a transposition table, which is a cache to store previously discovered states. This optimization would reduce the search space and allow for deeper search. The corresponding implementation in bitboards is known as *Zobrist hashing*.

The effects of a stronger evaluation function would also be interesting to research further. It would be especially interesting to develop an evaluation function using deep neural networks trained through self-play.

MCTS without cutoff and the evaluation function simulates playouts which play until the end of a game. This has been tested and as discussed earlier, the randomly played games heavily favored the foxes. To solve this issue, the evaluation and cutoff were implemented. However, a different approach could yield a different result such as implementing other algorithms which add heuristics to generate semi-random moves during the playout phase.

Another node expansion strategy could also be tested. In [5], it is stated that expanding one node at a time is often better than expanding all at once. Implementing this node expansion strategy could strengthen MCTS' play.

When it comes to the Fox Game, several game states could be studied more in depth such as states with few pieces left on board and common early game states which seem to often lead to temporary stalemates. One example of the stalemate states can be seen in figure 11, which is a position that often leads to stalemate no matter which player's turn it is. How knowledge of these types of positions affect the performance of the algorithms would be interesting to look into.

It could also be interesting to look for rule-set changes to make the game more fun and fair. For instance, the current

$$\begin{bmatrix} & & \emptyset & O & \emptyset & & \\ & & O & \emptyset & O & & \\ S & S & F & O & F & S & S \\ O & S & S & S & S & S & O \\ S & O & S & S & S & O & S \\ & & S & S & S & & \\ & & S & S & S & & \end{bmatrix}$$

Fig. 11. An early game position often leading to stalemate.

implementation of a turn limit makes it easy for the sheep to force draws.

VII. CONCLUSION

Two AI for the asymmetrical game the Fox Game were developed using MCTS and Alpha-Beta. The asymmetrical property favored the sheep as they are able to force draws in undesirable game states. In turn, a majority of the wins in the AI versus AI experiments belong to the sheep.

For the experiments, Alpha-Beta outperformed MCTS by a large margin in average win rate and is therefore more suitable for designing a simple AI for the Fox Game. However, there is still room for improvements for both algorithms. However, more research is required to better evaluate which algorithm is better at a more optimized state.

Results also showed a dominant victory for MCTS against humans, with only two losses out of ten matches. We estimate that MCTS has the playing power of a strong human player, which implies Alpha-Beta to be at least similar or better.

ACKNOWLEDGMENT

The authors would like to thank our supervisors Mika Cohen and Farzad Kamrani for their continuous support and guidance throughout the project.

REFERENCES

- [1] J. Mycielski, *Chapter 3 Games with perfect information*, ser. Handbook of Game Theory with Economic Applications. Amsterdam: Elsevier, 1992, vol. 1.
- [2] D. Silver, T. Hubert, J. Schrittwieser, and D. Hassabis. (2018, Dec.) Alphazero: Shedding new light on chess, shogi, and go. DeepMind. [Online]. Available: <https://www.deepmind.com/blog/alphazero-shedding-new-light-on-chess-shogi-and-go>
- [3] (2022, Apr.) Rävspel. Nationalencyklopedin. [Online]. Available: <http://www.ne.se/uppslagsverk/encyklopedi/l%C3%A5ng/r%C3%A4vspel>
- [4] D. Poole, A. Mackworth, and R. Goebel, *Computational Intelligence*. Oxford: Oxford University Press, 1998.
- [5] M. H. Winands, "Monte-carlo tree search in board games," in *Handbook of Digital Games and Entertainment Technologies*. Singapore: Springer, 2017, pp. 47–76.
- [6] L. Kocsis and C. Szepesvári, "Bandit based monte-carlo planning," in *European conference on machine learning*. Berlin: Springer, 2006, pp. 282–293.
- [7] J. M. Tanur, "Margin of error," in *International Encyclopedia of Statistical Science*. Berlin: Springer, 2011, pp. 765–765.

Monte-Carlo Tree Search for Fox Game

Anton Janshagen and Olof Mattsson

Abstract—This report explores if Monte-Carlo Tree Search (MCTS) can perform well in Fox Game, a classic Scandinavian strategy game. MCTS is implemented using a cutoff in the simulation phase. The game state is then evaluated using a heuristic function that is formulated using theoretical arguments from its chess counterpart. MCTS is shown to perform on the same level as highly experienced human players using limited computational resources. The method is used to explore how the imbalance in Fox Game (favoring sheep) can be mended by reducing the number of sheep pieces from 20 to 18.

Sammanfattning—I denna rapport undersöks om Monte-Carlo trädsökning (MCTS) kan prestera väl i rävspel, ett klassiskt skandinaviskt strategispel. MCTS implementeras med en cutoff i simuleringsfasen. Speltillståndet utvärderas där med hjälp av en heuristisk funktion som formuleras med hjälp av teoretiska argument från dess motsvarighet i schack. MCTS med endast begränsade beräkningsresurser visas kunna prestera på samma nivå som mycket erfarna människor. Metoden används för att utforska hur obalansen i rävspel (som gynnar får) kan förbättras genom att minska antalet fårpjäser från 20 till 18.

Index Terms—Monte-Carlo Tree Search, Artificial Intelligence, Fox Game, Cutoff, Heuristic Function

Supervisors: *Mika Cohen & Farzad Kamrani*

TRITA number: *TRITA-EECS-EX-2022:182*

I. INTRODUCTION

The interest in artificial intelligence (AI) for board games has grown considerably in recent years. Monte-Carlo Tree Search (MCTS) specifically is a search algorithm that has gained much popularity in making intelligent bots for board games. In 2016 Deepminds AI AlphaGo, which combines MCTS and neural networks, was the first AI to beat a professional Go-player, which had been a longstanding goal in AI development [1]. Since then, similar combinations of MCTS and neural networks have achieved super-human strength in other demanding board games such as Hex, Shogi and chess.

The goal of this project was to examine if, and if so how, MCTS can be used to create a strong AI for Fox Game, a game that currently lacks one. Fox Game is a perfect information board game that, despite its simplicity, contains sophisticated strategy. It is similar to Chinese checkers and "Fox and Geese" in regards to both rules and game board.

II. FOX GAME

Fox Game (Swedish: Rävspel or Svälta räv) is an old Scandinavian game that is a mixture of many different games. It has also evolved into different variants over the years. One effect of this is that there are several different sets of rules for the game, and since the game is not as popular anymore there is no consensus on which are the best. The game is said to have roots back to the 15th century, but it is unclear how similar

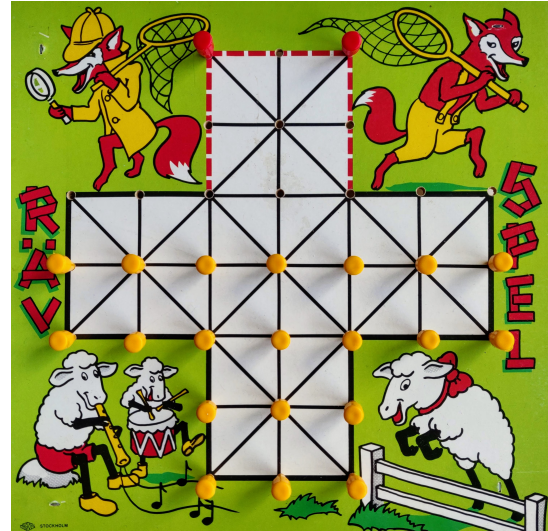


Fig. 1. Modern adaptation from Magtoys (photo: Anton Janshagen)

those versions were to the modern ones. In figure 1 a modern adaptation of Fox Game can be seen, and figure 2 shows the game board used in this report in its starting position.

The following rules are the same in all variants of Fox Game, and variations of them will be discussed below. The game is a turn-based two player game where one player plays as 20 sheep (purple pieces) and one player plays as 2 foxes (red pieces). The goal of the sheep is to move 9 sheep into the pasture (marked in green) on the other side of the board. The foxes do not have a goal of their own, but they simply try to stop the sheep from achieving their goal. The sheep may only move one tile at a time either side to side or forward, but not diagonally or backwards. The foxes however may move one tile at a time along all lines on the board.

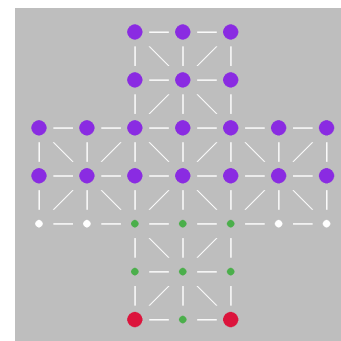


Fig. 2. Our implementation of Fox Game in its starting position

The foxes can capture sheep and thereby remove them from the board. This can be done when a fox is standing next to a sheep, and the space on the other side of the sheep is empty.

The fox may then move to the other side of the sheep while simultaneously removing the sheep from the board. If the fox is then put in a position where it may capture another sheep, it can immediately move and capture it without having to wait for the sheep to make a move first. See figure 3 for reference. The foxes cannot jump over each other.

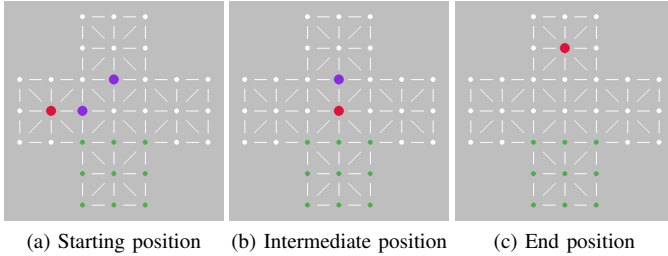


Fig. 3. Fox capturing two sheep in one turn

As mentioned above, there are certain alterations of the rules that deal with special situations that may arise when playing Fox Game. The following two rules can be found in many rule-books online but are not present in all of them. However we have included them in this report for reasons described below.

The first one is that the foxes not only may jump over the sheep to capture them, but must do so if presented with the opportunity, even if the fox has already jumped once this turn. The purpose of this rule is to give the sheep the possibility to sacrifice a sheep in order to lure a fox from, or into, a certain position. Particularly this stops the foxes from staying in the pasture and physically stopping the sheep from entering, as the sheep can force the fox to move away. One instance of this scenario can be seen in figure 4.

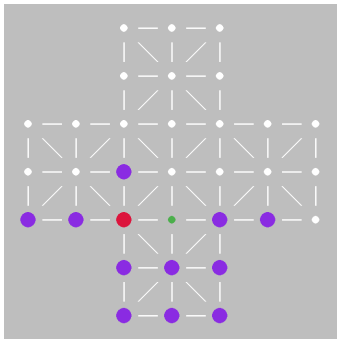


Fig. 4. The fox is forced to jump out of the pasture which allows the sheep to win

Another additional rule described in [2] is that not only the sheep can be captured, but also the foxes. The foxes are not captured in the same way as the sheep. They are instead captured if, on their turn, they have no possible move. So if a fox has no available move on the start of its turn it is removed from the board. This rule is necessary as the foxes otherwise can stay in the pasture to block the sheep from entering and thereby never lose. A fox being captured can be seen in figure 5.

A situation that is implicitly clear when humans play, but needs to be specified when bots play, is how to interpret the

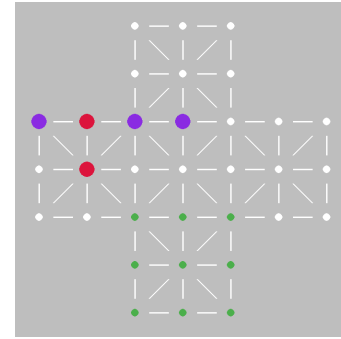


Fig. 5. The upper fox cannot move and is thereby captured

situation where the sheep cannot reach the pasture, but the foxes are not able to capture the sheep either. If for example the situation in figure 6 arises, the sheep has no reasonable chance to win but they can stay back forever and never lose. Our solution to this was to set a limit of how many turns in a row the sheep may move sideways. If the sheep have not moved forward after a set number of turns they lose. This is a reasonable interpretation of the rules as the foxes have successfully stopped the sheep from reaching the pasture. Throughout the report this number of turns was set to 10.

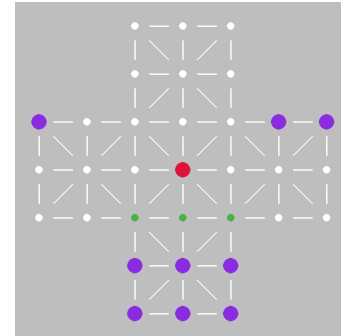


Fig. 6. The sheep can prolong the game inevitably

III. MONTE-CARLO TREE SEARCH

MCTS is a best-first tree search algorithm [3]. The algorithm creates a search tree where it evaluates different nodes in the tree based on Monte-Carlo simulations, and gradually moves further down the tree to get more and more precise evaluations of the different states.

The first node of the tree is the root node. It represents the current game state and is the only node without a parent node. The children of each node represent all of the legal moves that can be made from that board state. Beside what the board state looks like, all nodes contain information about their evaluation, the number of times they have been visited, which player is to make the next move, as well as pointers to their parent and children nodes.

MCTS consists of four main steps which can be seen in figure 7. The first step is to select from which node, i.e. which game state, the next evaluation should be made, this is called tree traversal. The process is done recursively from the root

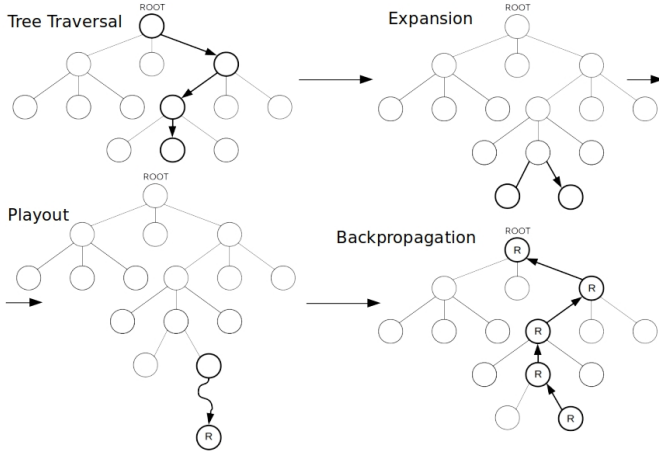


Fig. 7. Overview of the Monte-Carlo Tree Search algorithm

node until a node without children is reached. The selection between a nodes' children is done by picking the child with the highest UCB1-value according to equation (1), where Q_i is the value of the node, v_i is the number of times the node has been visited and v_{pi} is the number of visits of the parent node [3]. The first term is the average value of the node per visit and is called the exploitation term, and the second term is responsible for widening the search tree and is called the exploration term. C is a constant that is chosen to balance the depth and width of the search tree. The optimal value of C will depend on the branching factor of the search tree and therefore vary between different games. It can be seen from the UCB1 formula that a node with zero visits will have infinite UCB1-value and therefore always be chosen. If several children have infinite UCB1-value, one will be picked randomly.

$$UCB1_i = \frac{Q_i}{v_i} + C \sqrt{\frac{\ln(v_{pi})}{v_i}} \quad (1)$$

When a leaf node has been reached in the tree traversal phase the expansion phase starts. In this phase all of the possible child nodes are created, and a node is chosen at random.

The third phase is called playout. Here random moves are made from the game state associated with the leaf node until the game ends. The result of the game is one for a win and minus one for a loss.

The last part of the algorithm is backpropagation. The result from the playout is propagated back through the tree and each visited nodes' value and amount of visits is updated. Since every node is associated with a player, the value of a node must be updated based on if that player won the playout or not.

The process above is then repeated a certain number of times, known as the number of simulations. The more simulations you allow the algorithm, the more precise the evaluation will be. MCTS will not solve a game completely, but it will converge to the best solution given enough simulations [3]. The process can also be run for a specific amount of time, and the number of simulations will then vary depending on

how long time a simulation takes.

After a desired number of simulations has been done, or after a certain amount of time, the most promising child node of the root is chosen as the move to make. This can be done in two different ways, either by picking the child of the root with highest average value per visit, or by picking the child with the most number of visits. Since the child with the highest average value will have many visits, according to equation (1), these approaches yield the same choice in most cases. In this report the latter approach was used.

A. Cutoff

The idea of MCTS is that the nodes' values can be assessed with random simulations since a better position will lead to wins more often than a worse position when making random playouts from both positions. However this difference becomes smaller the more moves are made in the playout, because the difference drowns in the large randomness of the long playout. It can therefore be beneficial to introduce a cutoff in the playout phase after a certain number of random moves has been made. The game state can then be evaluated using a heuristic function. This can increase the performance of MCTS in games that require a large number of moves before either side wins, but it introduces the need for a heuristic function.

B. Heuristic Function

A heuristic function for a game can be made in different ways and include many different metrics. The output of the heuristic function should represent the probability of victory based on the current game state.

One common and traditional way to design a heuristic function is to consider both the value of the pieces left for each player, as well as where those pieces are. The value of the remaining pieces is called the material part, and the value of the pieces' positions is called the positional part. An example of this can be read about in [4], where a heuristic function is made for chess. The evaluation compares, for example, how many pawns are left for each player, while also considering that defended pawns are worth more than those who are easily captured by the opponent.

A common dilemma with heuristic functions discussed in [3] is that more complex functions may give a more accurate estimation of the board state, but can as a trade off be more computationally expensive. This means that fewer simulations can be made in the same amount of time, and therefore they may produce worse results even if the evaluation from the heuristic function is better.

IV. METHOD

All code was written in *Python 3.8* using the libraries *numpy* and *pygame* for calculations and visualization respectively.

A. Heuristic Function for Fox Game

The heuristic function used in this report took inspiration from its chess counterpart in [4]. The material

part consists of a weighted sum of the difference between the number of pieces of each sort for each player: $H_{mat} = w_f(F_1 - F_2) + w_s(S_1 - S_2)$. But since the players have zero pieces of one type the function simplifies to $H_{mat} = w_f F - w_s S$, where F is the number of foxes and S is the number of sheep. But we are really only interested in the relative value of the pieces, i.e. $\frac{w_f}{w_s}$. So one simplification that can be made is to set $w_s = 1$ and $w_f = q$, where q is how many sheep one fox is worth. This however means that the magnitude of the material part depends heavily on q , but the output should be a probability and thus be less than one. This was fixed by normalizing the coefficients with $(w_s + w_f)$, i.e. $(1 + q)$.

The positional part of the heuristic function used a piece-square table (PST) that assigns a value to each square. For each square that is occupied by a sheep, the associated value according to the matrix in (2) is summed to a variable V , which is used in the heuristic function seen in equation (3). The positional function only considered the sheep because observations showed that they needed help to know in which direction to move. Also a reasonable PST for the foxes is less obvious, and we wanted to influence the strategies of the AI as little as possible. The values of the PST were chosen by us based on prior knowledge of the game, but the idea is to encourage the sheep to move towards their goal.

$$PST = \begin{bmatrix} - & - & 0 & 0 & 0 & - & - \\ - & - & 1 & 1 & 1 & - & - \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 5 & 5 & 5 & 4 & 4 \\ - & - & 6 & 6 & 6 & - & - \\ - & - & 7 & 7 & 7 & - & - \end{bmatrix} \quad (2)$$

The positional part was reduced by a factor 0.1 so that the material and positional part have the same order of magnitude. The heuristic function also contains a parameter k , between zero and one, that balances the material and positional parts relative to each other.

All the variables in the heuristic function (F , S and V) were subtracted by their initial value to ensure that the function evaluates the game state to zero at the start of a match. The sum is also multiplied by a factor b that varies the overall magnitude of the function.

The heuristic function is also mapped to values between negative one and one with the hyperbolic tangent function (\tanh) to ensure that the result from a win or loss is worth more than a result from an evaluation by the heuristic function.

The heuristic equation can be seen in equation (3). It is stated from the perspective of the foxes, so it is close to one if the foxes are winning, and close to negative one if the sheep are winning. The evaluation for the sheep is the negative value of equation (3).

$$\begin{aligned} H &= \tanh[b(kH_{mat} + [1 - k]H_{pst})] \\ H_{mat} &= \frac{q}{1+q}(F - 2) - \frac{1}{1+q}(S - 20) \\ H_{pst} &= -0.1(V - 38) \end{aligned} \quad (3)$$

V. EXPERIMENTS

As a game lasts about 200 to 250 moves (when AIs play) a basic version of MCTS, without cutoff, would not be reasonable to use in the experiments. All experiments therefore used cutoff and a heuristic function. Most of the experiments were performed by letting agents with different constant-values play against each other. The agents played the same number of matches with both pieces because, as can be seen in the results, the sheep are heavily favored.

The experiments considered how altering constant-values in the heuristic function changed the strength of the agents, and how the number of simulations and cutoff-value affected their performance. The AI was then tested against human players, and an attempt to balance the game was made.

A. Experiment: Heuristic Function

The first experiment was to see how altering the different values (k , q , and b) in the heuristic function (3), and C in the UCB1 formula (1), affected the strength of an agent. A base agent with constant-values based on prior knowledge of the game and of MCTS was formulated, and agents with slightly different values played against it. The base agent used constant-values according to equation (4). Every altered agent played against the base agent 60 times, 30 times a sheep and 30 times as foxes.

$$\begin{aligned} k &= 0.8 \\ q &= 12 \\ b &= 1 \\ C &= 0.4 \end{aligned} \quad (4)$$

B. Experiment: Simulations

For the next experiment we tested how the number of simulations affected the performance of the agent. This was done partly to demonstrate that the algorithm works as intended, as a greater number of simulations should increase the performance, and partly to find how its performance depends on the number of simulations. This was done in order to suggest a number of simulations that balances performance and time consumption.

The experiment was done by having a base agent, according to equation (4), with 1000 simulations play at least 30 matches against agents with a different number of simulations for each set of 30 matches, and recording the results. All agents used the same heuristic function as the base agent.

C. Experiment: Cutoff

The benefits of the playout phase is that the added randomness helps the algorithm find strategies that at first might seem inferior. But without a cutoff, or too high cutoff-value, too much randomness will make the algorithm worse. The playout is also the most computationally demanding phase of the algorithm. So the cutoff-value has to balance both the amount of randomness in the playout phase, as well as how much time is spent in the playout phase versus the other

phases. It is possible that cutoff-value zero, i.e. immediate evaluation, is optimal because of the computations needed in the payout.

The cutoff experiments were divided into two parts. The first part used the base agent with a constant number of simulations (3000), but with different cutoff-values between different sets of matches. More than 30 matches were played between different agents, until an optimal cutoff-value was obtained.

This first part considered only how much randomness to introduce into the algorithm, while the second part focused on the time aspect of the cutoff. Here, the agents had the same amount of time to reach a decision, so the number of simulations had to vary accordingly. In the first experiment, with cutoff-value zero and five, the agents had 1.5 seconds, which amounted to about 4500 and 1100 simulations respectively. And in the second experiment, with cutoff-value zero and two, they had 1 second, which allowed for about 3000 and 1200 simulations respectively. In this second part of the experiment at least 50 matches were played between each pair of agents.

The goal was still to find an optimal cutoff-value, but only cutoff-values less than the value found in part one was considered. This is because if a higher cutoff-value is worse than a lower one when the same number of simulations is used, it will definitely be worse when the same amount of time is given. However this is not necessarily true for a cutoff-value less than the optimal value found in part one. This is because a lower cutoff-value allows for more simulations, which might be more important than having the optimal cutoff-value, if they are given the same amount of time.

D. Experiment: Strength Versus Humans

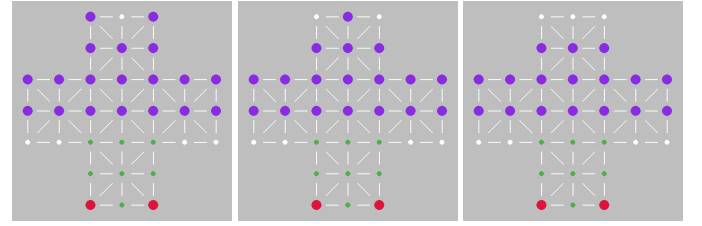
To truly see the strength of the agent we let it play against humans. The agent that played against humans was the base agent from equation (4) with the optimal cutoff-value from section V-C. The agent used 10000 simulations which gave it roughly three seconds to simulate. All human participants had prior experience of Fox Game. But since it is not a popular game, we never found anyone with more experience than us to face the AI.

E. Experiment: Imbalance of Fox Game

As Fox Game is an asymmetrical game it is not certain that the foxes and the sheep have equal chance of winning. From the data gathered in the heuristic function experiment V-A and cutoff experiment V-C, that both use high quality agents with many simulations, it can be seen that the sheep win 85% of games. With the current implementation of the rules Fox Game is a very unbalanced game. It is therefore interesting to examine how the game can become more balanced. As all of the variations of the rules, described in section II, plays an important role, it would be undesirable to tamper with them.

Our proposed solution is to reduce the number of sheep in the beginning of the game. To test how many sheep is appropriate to remove 30 matches were played between two copies of the best agent from the previous experiments with

10000 simulations (equivalent to about three seconds), but with different numbers of sheep at the start of the game. The heuristic function (3) was however altered to subtract the current amount of sheep and starting PST-value. So the factor $(S - 20)$ in H_{mat} was altered to $(S - S_{start})$, and the factor $(V - 38)$ in H_{pst} was altered to $(V - V_{start})$. One sheep was removed at a time from the row furthest from the pasture until the win rate started to favor the foxes. The sheep were removed in the pattern shown in figure 8.



(a) One sheep removed (b) Two sheep removed (c) Three sheep removed
Fig. 8. Starting positions in the Imbalance of Fox Game experiment

VI. RESULTS

The results that follow consists of approximately 1000 AI-played games, where some of them were played against humans.

A. Heuristic Function

When the base agent ($k = 0.8$, $q = 12$, $b = 1$ and $C = 0.4$) played against agents with slight variations that affected the heuristic function the win rates in table I were obtained.

TABLE I
RESULTS FROM AGENTS WITH DIFFERENT CONSTANT-VALUES PLAYING AGAINST THE BASE AGENT

Alteration	C = 0.5	C = 0.3	q = 14	q = 10
Base agent foxes wins	6	6	4	3
Base agent sheep wins	24	23	27	27
New agent foxes wins	5	7	3	3
New agent sheep wins	25	24	26	27
Base agent win rate	0.50	0.48	0.50	0.53

Alteration	k = 0.9	k = 0.7	b = 1.2	b = 0.8
Base agent foxes wins	8	11	3	2
Base agent sheep wins	24	26	25	25
New agent foxes wins	6	4	4	5
New agent sheep wins	22	19	28	28
Base agent win rate	0.61	0.50	0.47	0.45

B. Simulations

The win rate of agents with different numbers of simulations when playing against an opponent with 1000 simulations can be seen in figure 9. The win rate increases rapidly for agents with a low number of simulations, and for agents with a larger number of simulations the payoff in performance diminishes, this was an expected result.

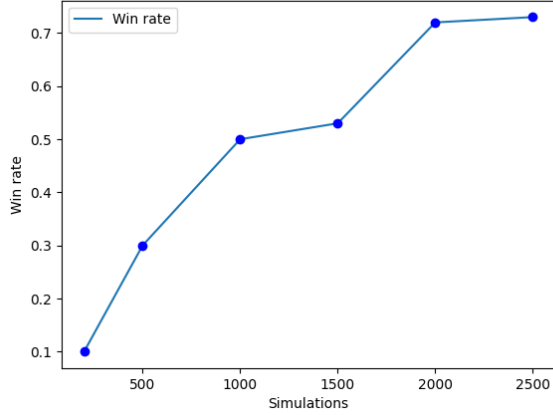


Fig. 9. Win rate against agent with 1000 simulations for agents with different number of simulations

C. Cutoff

In table II the win rates from part one and part two of the cutoff experiment can be found. In part one both agents used 3000 simulations, and in part two both agents had 1 or 1.5 seconds each to choose a move.

TABLE II
WIN RATES OF AGENTS WITH DIFFERENT CUTOFFS

Limiting resource Cutoff: Agent 1 - Agent 2	3000 Simulations		
	0 - 5	5 - 10	10 - 20
Agent 1 foxes wins	0	3	0
Agent 1 sheep wins	11	13	15
Agent 2 foxes wins	8	2	0
Agent 2 sheep wins	19	12	15
Win rate agent 1	0.29	0.53	0.50

Limiting resource Cutoff: Agent 1 - Agent 2	1.5 Seconds	1 Second
	0-5	0 - 2
Agent 1 foxes wins	6	8
Agent 1 sheep wins	23	22
Agent 2 foxes wins	2	8
Agent 2 sheep wins	19	22
Win rate agent 1	0.58	0.5

D. Playing Against Humans

All human players with limited experience of Fox Game lost every game, regardless of whether they played as sheep or as foxes. We, as more experienced players, also lost every game when we played as foxes, but as sheep we have managed to win several games. At the moment of writing we estimate that we can win every game when playing as sheep.

E. Imbalance of Fox Game

The win rate of the sheep decreases rapidly as sheep are removed from the game at the start. This shows how punishing it is for sheep to lose a piece without gaining terrain in return. But it also makes it hard to find an appropriate number of sheep to start with. As can be seen in figure 10, 18 sheep is the number of sheep that has a win rate closest to 50%. However

this number would only be suitable for skilled players. For novice players 20 sheep is probably better because of how hard it is to play as sheep.

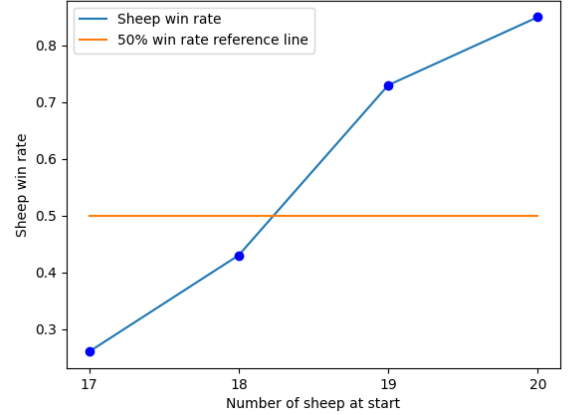


Fig. 10. Sheep win rate with different number of sheep at start of game

VII. DISCUSSION

Several discoveries were made from the experiments regarding both Fox Game and MCTS.

A. Strategies for Fox Game

As our initial knowledge of Fox Game and its different strategies was quite limited, a number of new and interesting discoveries were made during this project. Our first impression when playing the game against each other, as well as several testimonies online, suggested that the foxes are heavily favored. This is also what we found when the AI had a low number of simulations. But the results clearly show that when more advanced players, i.e. our AI with many simulations, play, the sheep are as heavily favored. As we played more against each other, as well as against the AI, we also performed substantially better as sheep than as foxes.

One crucial tactic that made the AI considerably better than most humans, which we never managed to master, was to reliably capture one or both foxes. It does this by thinking many moves ahead and by sacrificing some sheep in order to force the foxes into a position such that one of them has no available moves, and is thereby captured.

One interesting outcome of this is that the foxes often try to counter this strategy by staying further back. This however might not be optimal when playing against human opponents that are not as skilled in the strategy of capturing the foxes. It could therefore be the case that the optimal agent for beating an AI opponent is not the same as the optimal agent for beating a human opponent.

Observations of the AI:s play suggest that it does not have an optimal strategy, but it has impeccable tactics. That is, it seems to play optimally in every given situation, but lacks a plan that lasts more than a couple of turns. It can for example not sense the long term consequences of letting a fox through

its back line. This is in contrast to our strategy, that can beat the AI, in which we have a plan that lasts throughout the game.

This is most likely due to the positional part of the heuristic function being very simple and not valuing complex formations, and how spread out or clumped together the sheep are. While we try to focus on the entire board, the AI seems to focus more locally around the area where the foxes, and thereby the action, is. This is something that a more complex heuristic function that uses a neural network could do better. The authors of [5] used the neural network approach with great success in the strongest Go-playing AI to date.

B. Optimal Agent for an Imbalanced Game

The heuristic function and cutoff experiments that compared different agents gave vague results that indicated that most of the alterations between agents did not affect the strength of the agents very much. This can partly be derived from the imbalance of the game. Throughout these experiments the sheep won 85% of the games. This meant that a strong agent playing as foxes would still have a very low win rate against a slightly weaker agent playing as sheep. It also meant that a strong agent playing as sheep does not have the same possibility to further increase its win rate, as the win rate of the sheep is naturally very close to 100%.

It would therefore be easier to find differences between agents playing a more balanced version of the game, for example the one proposed in section VI-E. It is however not certain that the same constant-values of the agents are optimal for the balanced and imbalanced game. So one can not find an optimal agent for the balanced game, and suggest that that agent is optimal for the original game as well.

An interesting aspect about Fox Game being asymmetrical is that it is not certain that the constant-values of the agent that is optimal for playing as sheep, are the same as the constant-values that are optimal for playing as foxes. It could for example be more beneficial for the foxes to search wider in the tree (which would require a larger constant C) compared to the sheep. It would therefore be interesting to examine if different constant-values of the agents are optimal for playing as sheep, compared to playing as foxes.

C. Optimal Cutoff

It is interesting to note the significant performance increase of the AI when a cutoff was introduced into the MCTS algorithm. Prior to that, the agents played terribly and made huge mistakes, which resulted in the foxes winning almost every game. Our reasoning for this difference in performance is that Fox Game is unusually unforgiving relative to similar games. The sheep dictates the pace of the game completely and the foxes cannot force the sheep to do anything. If the sheep keeps control of the game they are heavily favored, but it is often enough with one or two minor mistakes from the sheep for the foxes to take control of the game, and there are a lot of bad moves for the sheep to make. So it might be that the punishing nature of Fox Game also punishes the randomness in the MCTS ployout where the sheep are bound to make big mistakes.

Another reason for the performance increase when introducing cutoff might be that the ployouts lasts quite long (200-250 moves), and approximately the same game state arises several times during the same ployout. This is because the sheep can move to the sides and then back again, and because if a few sheep in the back move to the side, the game state is functionally the same as before in many cases. This makes it hard to distinguish between a better and a worse starting position of the ployout.

The experiments to find the optimal cutoff-value in section V-C yielded some noteworthy results. They show that while doing random ployouts it is beneficial to use five as cutoff-value when both agents use the same number of simulations. Part two of the experiment however shows that the extra time spent doing the random ployouts is not worth it, as a cutoff-value of zero outperforms a cutoff-value of five when given the same amount of time rather than the same number of simulations. This is a very interesting result as the creators of the world's foremost AI:s for playing Go, AlphaGo Zero, have come to the same conclusion for their AI [5].

VIII. CONCLUSION

As seen in the experiments above a strong MCTS-agent for Fox Game requires a very low (or zero) cutoff-value, and therefore also a heuristic function. They also show that Fox Game is a very unbalanced game where the sheep win a majority of games if both players are playing intelligently. This makes the development of a strong AI difficult because the difference in strength between two agents is hard to distinguish without huge amounts of data. We have also seen that our best performing AI outplays humans with limited experience, and does well against more experienced players.

ACKNOWLEDGMENT

The authors would like to thank our supervisors Mika Cohen and Farzad Kamrani for their guidance and investment throughout the project.

REFERENCES

- [1] DeepMind. (2022, Apr.) Alphago. [Online]. Available: <https://deepmind.com/research/case-studies/alphago-the-story-so-far>
- [2] SpelRegler. (2022, Mar.) Rävspelet regler. [Online]. Available: <https://www.spelregler.org/ravspelet-regler/>
- [3] M. H. M. Winands, "Monte-Carlo tree search in board games," in *Handbook of Digital Games and Entertainment Technologies*, P. C. Ryohei Nakatsu, Matthias Rauterberg, Ed. Singapore: Springer Singapore, 2017, pp. 47–74.
- [4] C. E. Shannon, "Xxii. programming a computer for playing chess," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 41, no. 314, pp. 256–275, 1950. [Online]. Available: <https://doi.org/10.1080/14786445008521796>
- [5] 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 (London)*, vol. 529, no. 7587, pp. 484–489, 2016.

CONTEXT R

BIG GRAPHS IN SOFTWARE DEVELOPMENT

POPULAR DESCRIPTION

Developer deletes 11 lines of code – the consequences will shock you!

The majority of applications which most people take for granted are dependent on the contributions of a vast network of developers who freely share their software under Open Source licenses. This practice has greatly increased the rate of technological development. Instead of constantly reinventing the wheel developers are instead able to reuse established solutions to reoccurring problems. The reuse of Open Source code is facilitated through package managers, which can be viewed as ecosystems of software. Although practical, if unchecked, these interconnected software ecosystems can become a digital card house that collapses.

When developer encounters a problem he is unsure of how to solve, a common practice is to find an existing package which solves this problem for them. If he is lucky, someone else has encountered the same problem, and uploaded the solution to one of the many existing online repositories (collections) of open source packages. All the developer now has to do is download the package, and list it as a dependency to his own project source code. Problem solved.

However, things are unfortunately not always so easy. In fact they can get horribly wrong, as the following example shows. A common problem for a number of web developers was inserting space to the left side of objects in their graphical applications. Enter 'leftpad' was a javascript package providing the functionality of adding space to the left side of objects. It was written by one developer. When he removed it from the node package manager, the internet broke although 'leftpad' only consisted of 11 lines of code. Thousands of applications were unable to compile or run due to the missing dependency. This even affected applications, which had no explicit reference to 'leftpad'.

The reason the fallout of this deleted package was so vast, and how developers who had no knowledge of the package were also affected, is because when an application is compiled it does not only download its own dependencies, but also any dependency listed in the packages they depend on. And all the dependencies those packages depend on. And all the dependencies those packages depend on. Quite quickly, a single package can be connected to a majority of packages in a software repository, and when it is removed, the whole ecosystem comes with it.

The study of big graphs in software development is concerning software ecosystems, which are package repositories for specific programming languages where developers can submit and download open source software. The packages in these ecosystems are connected to each other explicitly through their direct dependencies and indirectly through transient dependencies (dependencies of dependencies). The aim of studying these graphs is both to ease the maintenance of code for developers, as well as analyze trends that occur, so that incidents such as 'leftpad' do not occur again. Studying these graphs also allows to understand what makes certain packages more popular than others. Although the study of software ecosystems and dependency graphs existed before 2016 when the 'leftpad' incident occurred, it has definitely gained more attention, especially from developers themselves concerning their own projects.

SUMMARY PROJECT RESULTS

Big graphs in software development refers to the graphical representation of dependency trees in software ecosystems (package repositories for specific programming languages). Dependency trees are composed of software packages, represented as nodes, which are directionally connected to their dependent packages through edges. These graphs are dynamic due to new packages constantly being created and existing packages evolving to offer new or improved functionality.

These graphs are created to help humans visualize the increasingly complex software supply chain. This can help aid developers in making more informed decisions concerning their own projects when outsourcing functionality to a third party package through a dependency, as any vulnerability in the dependency will also be present in their own package. Also, these graphs can be used to study trends across entire software ecosystems and identify problems such as over reliance on a single package. The results of such studies can be used to incentivize entirely new packages to be developed in order to decouple the ecosystem.

In project R1 the software supply chain of the Ethereum blockchain was studied in detail. To prevent the entire blockchain being dependent on a single client to communicate with the network there exist several clients written in different languages in order to minimize the potential risks of a single client crashing. In the conducted study the dependency graphs of the most popular Ethereum clients were analysed for both quantitative and qualitative metrics. These metrics were compared across clients from within the same ecosystem, as well as from ecosystem to ecosystem. Earlier studies in this field have either focused on a single ecosystem, and analysed the evolution of software graphs over time, or looked at multiple ecosystems, but only compared static snapshots of software graphs. Concerning the Ethereum blockchain only one study has been conducted in which the evolution of the software supply chain of a single ecosystem was analysed. In project R1 a study was conducted where the software supply chain of multiple ecosystems was analysed and compared, which is a novel dataset.

Future studies in the same area as project R1 would be to extend the scope of the study to include all the ecosystems for which there is an Ethereum client, as well as looking at other software platforms which are distributed among different software ecosystems.

IMPACT ON SOCIETY AND SUSTAINABILITY

As essential services are increasingly becoming digital, software maintenance and service uptime is of great social concern with potential implications on public health, safety and finances. Big software graphs, and the study of the software supply chain, is a fairly young niche research topic. The aim is to inform software suppliers and consumers of the state of digital infrastructure. Although this topic has previously mostly concerned developers, it is also of great public interest.

Studies have shown that the majority of bugs in software are due to known vulnerabilities. The same goes for malware introduced by malicious actors wanting to exploit end users of software. An example of the societal effects of inaction regarding such known vulnerabilities is the global IT-attack in 2021 which forced all Coop grocery stores in Sweden to close for several days. This was due to malware being introduced not to the cash register system of Coop, but rather being introduced several steps down the software supply chain, which Coop themselves have limited control over. The financial impact of this has not been disclosed, but can be assumed to be very large. Luckily for citizens of Sweden there exist several chains of grocery stores relying on different cash register software, so the effects on consumers was limited, if none at all. This example highlights how big software graphs are of concern not only to developers of software to identify potential vulnerabilities and bugs, but also to end users of software so that they can make informed decisions regarding the services they rely on. The fact that citizens in Sweden were still able to buy groceries due to the existence of several chains of stores is also allegorical to the need for software diversity within software ecosystems. The consequence of this target attack is miniscule when compared to the potential effects of other essential digital services. As a response to the covid-19 pandemic, the entire vaccination strategy in Sweden was managed by a single application named AlltidÖppet, which translates to AlwaysOpen. The ability of this application to be what it is named has great implications on society in terms of health and finances. Any downtime of this application would reduce the efficiency of vaccination which would have dire consequences on public health.

The environmental impact of the study of software graphs is limited, although not null. A large concern of big software graphs is about version compatibility and about providing ways for developers to more easily maintain software to software compatibility. It can be argued that this work can also be used to more easily maintain software to hardware compatibility, which in turn could prolong the life cycle of hardware before they become obsolete due to the lack of software updates. As modern electronics require numerous rare earth metals, of which there is a finite supply, prolonging the life cycle of hardware would reduce the demand for mining, and thus have a positive impact on the environment.

The State of Software Diversity in the Software Supply Chain of Ethereum Clients

Noak Jönsson

Abstract—The software supply chain constitutes all the resources needed to produce a software product. A large part of this is the use of open-source software packages. Although the use of open-source software makes it easier for vast numbers of developers to create new products, they all become susceptible to the same bugs or malicious code introduced in components outside of their control. Ethereum is a vast open-source blockchain network that aims to replace several functionalities provided by centralized institutions. Several software clients are independently developed in different programming languages to maintain the stability and security of this decentralized model. In this report, the software supply chains of the most popular Ethereum clients are cataloged and analyzed. The dependency graphs of Ethereum clients developed in Go, Rust, and Java, are studied. These client are Geth, Prysm, OpenEthereum, Lighthouse, Besu, and Teku. To do so, their dependency graphs are transformed into a unified format. Quantitative metrics are used to depict the software supply chain of the blockchain. The results show a clear difference in the size of the software supply chain required for the execution layer and consensus layer of Ethereum. Varying degrees of software diversity are present in the studied ecosystem. For the Go clients, 97% of Geth dependencies also in the supply chain of Prysm. The Java clients Besu and Teku share 69% and 60% of their dependencies respectively. The Rust clients showing a much more notable amount of diversity, with only 43% and 35% of OpenEthereum and Lighthouse respective dependencies being shared.

Sammanfattning—Mjukvaruleverantörskedjan sammanfattar resurser som behövs för att producera en mjukvaruprodukt. En stor del av detta är användningen av öppen källkod. Trots att användningen av öppen källkod tillåter snabb produktion av nyprodukter, utsätter sig alla som använder den för potentiella buggar samt attacker som kan tillföras utanför deras kontroll. Ethereum är ett stort blockkedje nätverk baserat på öppen källkod som försöker konkurrera med tjänster som tidigare endast erbjudits av centraliserade institutioner. Det finns flera implementationer av mjukvaran som implementerar Ethereum som alla utvecklas oberoende av varandra i olika programmerings språk för att öka stabiliteten och säkerheten av den decentraliserade modellen. Idenna rapport studeras mjukvaruleverantörskedjorna av de mest populära klienterna som implementerar Ethereum. Dessa utvecklas i programmeringsspråken Go, Rust, och Java. De studerade klienterna är Geth, Prysm, OpenEthereum, Lighthouse, Besu, och Teku. För att genomföra studien transformeras klienternas mjukvaruleverantörskedjor till ett standardiserat format. Kvantitativt används för att beskriva dessa leverantörskedjor. Resultaten visar en stor skillnad i storlek av leverantörskedjor för olika lager i Ethereum. Det visas att det finns en varierande mångfald av mjukvara baserat på de språk som klienter är utvecklade med. Leverantörskedjorna av Go klienter sammanfaller i princip fullt, medan de av Java klienter sammanfaller med en stor majoritet, och de av Rust klienter visar på mest mångfald i mjukvarupaket

Index Terms—Software Supply Chain, Dependency Graphs, Open Source Software, Software Diversity, Ethereum, Blockchain

Supervisors: Benoit Baudry, César Soto-Valero

TRITA number: TRITA-EECS-EX-2022:183

I. INTRODUCTION

The software supply chain is comprised of all resources, human and technological, required to produce a software product [1]. A significant component of this software supply chain are package managers, which are programming language-specific collections of open-source software packages. Package managers distribute open-source packages through online repositories and provide methods of collecting all dependencies required for a certain package. These package managers are often referred to as ecosystems, as the packages they constitute are interconnected through dependencies. The use of open-source packages from these software ecosystems is not only limited to use for other open-source projects. Instead, it has been shown that 85% of the source code for an average enterprise application is from open source packages [2]. Although the reuse of open-source software packages can allow faster development of new projects, all dependent projects become susceptible to the same bugs that emerge in a dependency, as well as malicious code injections.

Software supply chain attacks are one of the most prevalent methods used by malicious actors in order to compromise software, and a growing concern for both developers and policy makers [3]. One common method is *Typosquatting*, where malicious actors release software packages with names that are slight spelling variations of popular open-source packages, hoping to trick developers into including these infected packages [4]. Malicious actors may also try to infect existing packages by gaining access to the repository where the source code is hosted either by social engineering or by hacking the account of someone who already has access.

The Ethereum blockchain is a distributed software platform built entirely using open-source software. Through the use of smart contracts, digital assets can be exchanged between the parties involved without the need for a centralized governing institution. These smart contracts can be written to provide functionalities such as financial services, digital art trade, online games.

There are exist several Ethereum clients, developed in different languages. A vast network of nodes, computers which run these clients, all communicate each other in order to host the Ethereum Virtual Machine. The clients are split into to groups; the execution layer (Eth1), which is responsible for appending new transactions to the blockchain, and the

consensus layer (Eth2), which is responsible for making sure the added transactions are distributed among all the nodes.

In this paper, we study the software supply chain of three pairs of Ethereum clients. The analysis is narrowed down to focus specifically on the software diversity of open source dependencies and their suppliers. The studied clients are GoEthereum and Prysm, developed in Go; Besu and Teku, developed in Java; and OpenEthereum and Lighthouse, developed in Rust. These clients are chosen for two reasons: 1) they include the most popular clients in use, combined they total over 90% of all nodes in the Eth1 execution and Eth2 consensus layer currently in use; 2) the existence of a client written in a particular language in both Ethereum layers.

The analysis was conducted by download and build each client from the the source code. Outputting the dependency trees of each client using their native package manager. Reformatting this output into a uniform format and finally performing analysis on each tree individually, and comparing trees from the same ecosystem.

In summary, this paper makes the following contributions:

- The notion Unified Dependency Tree as a way to study the diversity in the software supply chain of distinct Ethereum clients.
- Novel metrics regarding Ethereum Clients including: total dependencies, unique direct dependencies, unique transitive dependencies, and unique suppliers.
- Insights into the distribution of suppliers within different ecosystems which validates past research.

II. BACKGROUND

A. Software Supply Chain

Software supply chains are all the resources required to produce a software product. This includes human resources, such as developers, teams, and larger organizations. Technical procedures such as automatic testing and build processes [5]. Lastly, this also includes other software products such as standard libraries, tools, and third-party software packages. The focus of this paper will be on the software diversity in software supply chains with regard to third-party open-source software (OSS) packages and their suppliers.

The use of OSS by developers to create a new product is a cornerstone of modern development practices. In order to feasibly facilitate the reuse and distribution of OSS, developers often rely on package managers. Package managers are programming language-specific repositories of OSS packages [6]. Not only do they host the source code for OSS packages, but they also provide methods for downloading, updating, and building packages. Examples of package managers are Gradle and Maven for Java, PyPi for Python, and Cargo for Rust. Go, which is used to develop two of the clients studied in this report, does not utilize a package manager. Although Go does not have central repositories, the language does provide a tool for downloading and updating packages.

In order to utilize the functionality provided by an OSS package in a project, a developer must declare it as a dependency. Software dependencies are packages that are required

by another package in order to function. Declaration of dependencies is accomplished through the use of a file in the root of the project directory.

Listing 1 shows an example of how dependencies are listed for a project developed in the language Rust, utilizing the cargo package manager. Common for all package managers is to list the name of the package, which is unique. Most, although not all, package managers also require a specific version of the dependency package to be declared. Any package referenced explicitly as a dependency in a project is defined as a direct dependency on said project. As direct dependencies themselves may have their own dependencies, they are also dependencies to the project. These dependencies are defined as transient dependencies.

```
[package]
description = "OpenEthereum"
name = "openethereum"
# NOTE Make sure to update util/version/Cargo.toml
# as well
version = "3.3.4"
license = "GPL-3.0"
authors = [
    "OpenEthereum developers",
    "Parity Technologies <admin@parity.io>"
]

[dependencies]
blooms-db = { path = "crates/db/blooms-db" }
log = "0.4"
rustc-hex = "1.0"
docopt = "1.0"
clap = "2"
term_size = "0.3"
textwrap = "0.9"
num_cpus = "1.2"
```

Listing 1. Example of dependency declaration in Rust using Cargo.toml file.

B. Software Supply Chain Attacks

Software supply chain attacks are directed attempts to inject malicious code into a software package in order to compromise any and all software packages which are dependent on the targeted package. In 2021 the EU Agency for Cybersecurity reported that 66% of cyber attacks target the software supply chain [7]. Decan et al. [6] analyzed the trends in seven different software ecosystems. They found that a majority of software packages are dependent on a minority of core packages. This highlights how a successful and well-targeted attack can affect the majority of a software ecosystem.

Software supply chain attacks targeting the dependency tree can be split into two categories; those infecting existing packages and those that create new packages containing malicious code [4]. When infecting an existing package, culprits often rely on existing known vulnerabilities in the code. Otherwise, they need access to the project, which can be achieved through social engineering, i.e., manipulating their way to get maintainer privileges for the project or by gaining the credentials of a person who is a maintainer of the project. When creating new packages containing malicious code, the culprit must still inject the package into some software supply chain. This is most commonly achieved through Typosquatting, which is when a package given a name that is a slight spelling variation

from that of a popular package. For example a package could be named 'bloons-db' instead of 'blooms-db', as seen in Listing 1. Other ways of injecting a malicious package include creating a Trojan Horse, where a package claims to provide some functionality but has a built-in backdoor mechanism to allow culprits to extract data from the end-users of the project.

C. Software Diversity

Software diversity is a concept with a broad scope in the study of software development [8], [9]. In general, it refers to the existence of multiple software components which are functionally similar, but implemented and created in separate ways. The aim of software diversity is to encourage fault tolerance, security, and reusability [10].

This paper deals mostly with the concept of design diversity and managed natural diversity. Design diversity refers to the practice of independently developing multiple software projects according to the same specification. Utilizing these projects simultaneously yields a more fault tolerant system due to "the independence of failures among the diverse solutions" [10]. Managed natural diversity emerges as a result of development practices. As open source licenses give anyone the right to copy, modify, and redistribute an OSS packages, this practice has the potential to yield vast amounts of software diversity [11]. The opposite of software diversity is monoculture, where a single software supplier, or a single package, is heavily reoccurring in a software supply chain. Monoculture provides malicious actors a definite target from which a malicious code injection could have a tremendous impact.

D. Ethereum Ecosystem

Ethereum is an open-source decentralised software platform used for finance, digital art, and a host of web3 applications [12]. Based on blockchain technology, Ethereum functions by allowing users to share and trade digital assets through smart-contracts, which are recorded in a digital ledger. The contents of the digital ledger are maintained and agreed upon by a vast number of nodes, which are computers that support the Ethereum Virtual Machine (EVM).

The Ethereum Foundation promotes design diversity, in the form of client diversity [13]. There is no official implementation, rather there are several clients developed in different programming languages, as to increase software diversity by leveraging several ecosystems of OSS packages.

The function of the execution layer (Eth1) is to add new blocks of transactions to the shared state of the network. Eth1 uses a proof-of-work (PoW) mechanism in order to ensure that the new state is valid. When a transaction occurs, and is to be added to the blockchain, nodes running an Eth1 client compete against each other in completing a computationally heavy task. The first node to complete the task is allocated the block, and all other nodes with point to it as the correct state. The currently available Eth1 clients, the language they are developed in, and the percentage of nodes running them are shown in Table I.

The function of the consensus layer (Eth2) is to make sure that the updated state of a new block being added to the

Table I
EXECUTION LAYER (ETH1) CLIENTS

Client	Programming Language	Distribution
GoEthereum	Go	84.33%
Erigon	Go	7.26%
OpenEthereum	Rust	5.77%
Nethermind	C#	1.78%
Besu	Java	1.22%

Table II
CONSENSUS LAYER (ETH2) CLIENTS

Client	Programming Language	Distribution
Prysm	Go	38.34%
Lighthouse	Rust	33.51%
Teku	Java	16.51%
Nimbus	Nim	11.54%
Lodestar	Typescript	0.01%

chain is distributed amongst all the nodes in the network. Eth2 uses proof-of-stake (PoS) validation. This consensus method is more energy efficient, as no computationally heavy task is required. In this method, nodes stake their own capital as collateral in order to ensure that they behave correctly. The currently available Eth2 clients, the language they are developed in, and the percentage of nodes running them are shown in Table II.

As the Ethereum blockchain is used to provide functionalities such as cryptocurrencies and decentralized finances any vulnerabilities in its software supply chain can have dire economic consequences. Mitigating this risk involves both client diversity, as a critical bug in the Eth2 consensus layer of a client used by more than 33% could cause the blockchain to go offline, as well as diversity of dependencies across the Ethereum ecosystem so that multiple clients are all affected by the same bug or malicious code injection [13].

III. RELATED WORK

In 2021 the EU Agency for Cybersecurity reported that 66% of cyber attacks target the software supply chain, and provide an outline for software consumers to navigate the growing threats [7]. Following an Executive Order in 2021 to secure the Software Supply Chains in the USA, the Cybersecurity and Infrastructure Security Agency of USA released a paper with the aim to introduce standards for software supply chains to ensure their security [7].

In 2017, Decan et al. published an analysis comparing the evolution of dependency trees in 7 popular software ecosystems. Common trends among all studied ecosystems were that they all tend to grow in number of projects and dependencies. Also, in every ecosystem the majority of projects are dependant on a minority of core projects. Differences between ecosystems included the degree of transitive dependencies, where some ecosystems remained stable while others saw an increase in the ratio of transient dependencies to direct dependencies over time. The paper also introduced new quantitative metrics to determine the *health* of ecosystems [6].

One major contribution to the study of software supply chains and software ecosystems is the Maven Dependency

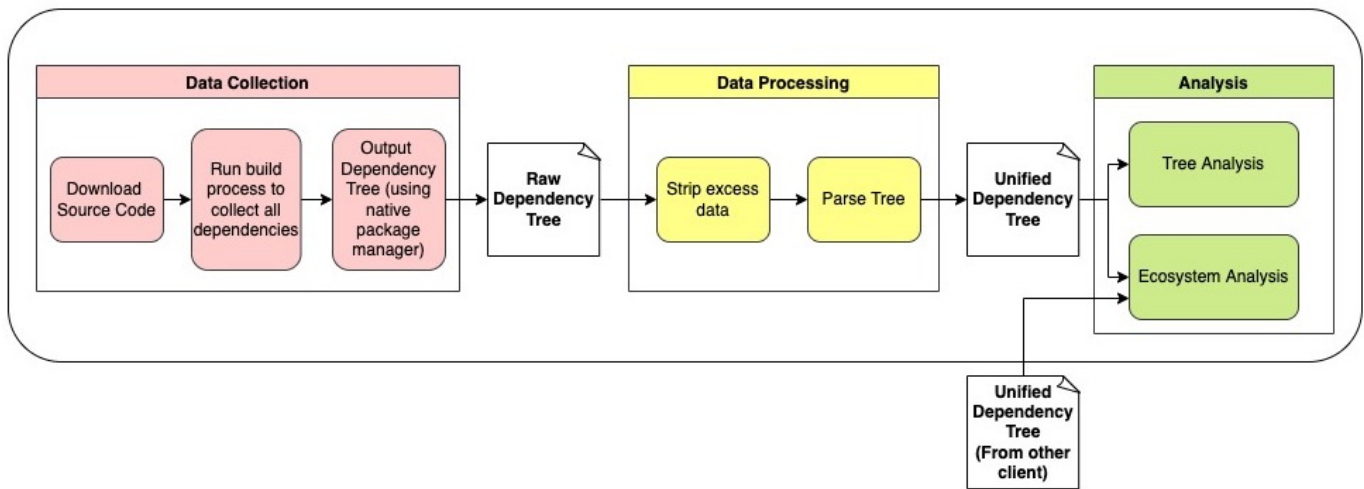


Figure 1. Data pipeline for the analysis of the software supply chain of a single Ethereum client.

Graph. Presented in a 2019 paper, the Maven Dependency Graph is a snapshot of the Maven Central Repository, a package manager for Java projects. The dependency graph contains 2,4 million artifacts and 9 million dependencies, stored in a graph database with an accompanying API for querying the data set [14].

Studying software supply chains in PyPI, the package manager for Python projects, Benthall et. al introduced a model for identifying *hot spots* of risk in ecosystems [15]. These *hot spots* are determined through static analysis and show projects that is highly connected to the rest of the ecosystem through reverse dependencies, and is exposed to a large number of vulnerabilities through their dependencies.

In 2020, Ohm et al. released a paper reviewing several malicious software packages, which have been used to attack software supply chains, and outlined the main methods used to inject malicious code into the supply chain. Most commonly malicious code is introduced through *typosquatting* [16], releasing a project with a name with a slight variation to that of a prominent package [4]. The second most common way is by infecting an established project, which requires the culprit to have gained access to the project either by taking over a maintainers credentials, or becoming a maintainer through social engineering.

Conducting a longitudinal analysis of Java projects from the Maven ecosystem Soto-Valero et al. focused on the usage status of dependencies in projects. They found that bloated dependencies, that is dependencies which are not actually used by the project, in most cases remain bloated [17]. Also, they showed that bloat tends to grow over time, even in cases where developers remove direct bloat. This shows that developers seldom encounter negative implications from removing currently bloated dependencies, and that all developers who maintain projects need to be diligent in pruning their supply chain for the entire ecosystem to improve.

Pashchenko et al. studied the supply chains of 200 popular Java projects, which totaled over 10000 distinct dependencies, in order to analyse the effect of software vulnerabilities. Their study showed that 20% of known vulnerabilities are

never deployed and do not pose a threat to the dependant project [18]. It was also found that 81% of vulnerabilities in a projects supply chain could be fixed by updating the vulnerable dependency to a newer version. Of the studied vulnerable dependencies 1% were halted, ie abandoned by their maintainers, and posed a severe problem for downstream projects.

In 2020 Zamani et al. conducted a review of 40 security breaches of the Ethereum blockchain. They found that vulnerabilities in clients was the 2nd most prominent cause of security incidents [19].

Aumasson et al. released a security review of the Ethereum beacon chain. They discuss that although bugs in dependencies required for cryptography procedures would have more severe outcomes to the performance of the network, "all dependencies execute code at the same privilege level" and therefore potentially harmful [20].

In 2022 Soto-Valero et al. released the first study into the software supply chain of Ethereum clients from the Java ecosystem, and looked at the evolution of the supply chains over a one year period. They found that both the number of dependencies and suppliers increased over this period. They also found that the majority of dependencies were shared amongst both clients supply chain. [21]

In a 2022 report Enck et al. summarized 3 summits they held with organisations representing both enterprise and USA policy makers discussing challenges in securing the software supply chain. [3]. A frequently recommended security measure which was discussed and disagreed upon was that of automatic version updates of dependencies. Although security experts maintain this practice eliminates exposure to vulnerabilities, many developers argued this often led to bugs and breaking changes being introduced to their projects due to immature code. It was also mentioned that there have been incidents where software projects had been infected with malicious code, as mentioned in [4], which shows that automatically updating is not a bullet proof strategy. Another point of discussion was the practice of providing a standardized Software Bill of Materials (SBOM).

IV. METHODOLOGY

A. Project Pipeline

In order to generate a data set depicting the software supply chain of the Ethereum ecosystem, each studied client was treated according to Figure 1 individually. For each client, the latest version of the source code was downloaded from their respective GitHub repository. The build process was then invoked in order to download all direct and transient dependencies. The client software was then run to ensure that all dependencies were fetched and that the software was functional. Using the native package manager of the client their dependency tree was output. From this step in the pipeline until the analysis of the unified dependency trees, the implementation of the procedure varied depending on the package managers used. In general terms the next step was to strip the raw dependency tree of data irrelevant to the study. Examples of irrelevant data include internal (non-third-party) dependencies as well as paths pointing to the location of dependency source files on the system. Next, the raw dependency tree was parsed. Individual packages were formatted according to the artifacts schema, and the dependency relationships were formatted according to the dependency schema, both defined in IV-B. This process differed depending on the format of the raw dependency trees, which were either nested trees or lists of package pairs. Once the data was structured in the unified dependency tree format, the same procedure for analysis was utilized for all clients. Individual trees were analysed in order to collect metrics defined in section IV-C. Unified Dependency Trees of clients developed in the same programming language were also analysed together in order to collect data regarding to the intersection of their dependencies. Details of differences in implementation, and technical difficulties, of clients are described below.

1) *Go*: Package management in Go differs from most other programming languages in that it does not utilize a third-party package manager, nor a central repository to host software packages. Where packages are hosted is instead left up to the supplier. From manual inspection of the dependencies in the studied Go clients, GitHub is the most common hosting solution. Go does not use differentiate dependencies by scope, rather all dependencies are compiled when build procedures are invoked. In order to define dependencies in a Go project, a developer lists each dependency by the URL which points to where the package is hosted followed by its version. This is done in a file named `go.mod` in the project directory. The command `go mod graph` will output a list of all dependencies in the project, including internal non-third-party dependencies, with each line containing a dependent package and its dependency separated by a space. In order to remove internal dependencies, the command `go list -m` is used to curate a list of third party packages. These lists are used together to ensure that the unified dependency tree only consists of third party dependencies.

2) *Rust - Cargo*: The Ethereum clients written in Rust all use the Cargo package manager. All dependencies for a project are listed in a file named `Cargo.lock`. The dependency tree for a Cargo project can be output using the `cargo`

`tree` command. The output of this command is a nested tree structure.

3) *Java - Gradle*: The Ethereum clients written in Java all use the Gradle package manager. Projects using gradle lists all dependencies in a file named `build.gradle`. The dependency tree is output using the command `gradle -q dependencies`. The output consists of several nested trees, separated by the scope of the dependencies. The output also includes non-third-party dependencies, however these were easily identified through machine-readable means and removed systematically.

B. Unified Dependency Tree

As seen in the end of section 1, the collection of data regarding a projects software supply chain is not trivial, and differs greatly between package managers. In order to facilitate easier analysis of supply chains across several software ecosystems a uniform format for this data is desirable. Also, as efforts are being made to introduce a the practice of providing software bill-of-materials (SBOM) in the software industry, a standardised model for supply chains is needed [3]. In this report, the notion of a Unified Dependency Tree is introduced. The model defines data structures using json format, as it is structured text-based format which is both human-readable and recognized by several scripting languages [22]. The model represents software packages using the data type *Artifact*. This structure has four key-value pairs which are, `artifactId`, which is the name of the package, `groupId`, the supplier of the package, `version`, and finally the `gav`, which is a concatenation of the first the values. The `gav` is used as the unique identifier for the artifact. All the dependencies for a project are stored in a json object. The unique id `gav` of each dependent artifact in the project is entered into this object and points to a list of the `gav` of all its dependency artifacts.

C. Metrics

For the analysis of individual clients the following metrics were collected

- Total dependencies
- Unique direct dependencies
- Unique transient dependencies
- Unique suppliers

Total dependencies is the sum of all dependencies. A package which is a dependency of several packages is counted once for each dependent package. If a dependent package is featured multiple times, its dependencies are only counted once for the dependent package.

Unique dependencies is the sum of all packages which the client is directly dependent upon as declared in their source code according to the methods described in section IV-A.

Unique transient dependencies are the sum of all packages which are featured in the dependency tree, but not directly dependent. Packages which are dependencies to several packages are only counted once.

Unique suppliers are the sum of all suppliers of packages featured in the dependency tree. Suppliers who provide more than one package, or who supply a package which is featured multiple times, are only counted once.

V. RESULTS

RQ1. What is in the supply chain of Ethereum Clients?

After analyzing all Ethereum clients individually, there is an evident size difference between the supply chains of the two Ethereum layers as shown in Table III. Besides Besu having more unique direct dependencies than Teku, the supply chain metrics gathered are much larger for the Eth2 clients compared to the Eth1 clients of the same ecosystem. The biggest difference is seen in the Go ecosystem, in which the metrics of the Eth2 client Prysm is at least twice the size of the metrics of the Eth1 client Geth.

All the Eth1 clients require roughly the same amount of unique direct dependencies, while having vastly different amounts of unique transitive dependencies. For Geth (Go), there are 3.2 unique transient dependencies per unique direct dependencies; for OpenEthereum (Rust) this ratio is 5.7; for Besu (Java) it is 3.0. Although the Eth2 clients have vastly differing amounts of unique direct dependencies, the ratio of unique direct dependencies to unique transient dependencies are similar to the Eth1 clients from the same ecosystem; Prysm (Go) 4.3; Lighthouse (Rust) 5.6; Teku (Java) 3.0.

There are no clear patterns between the number of suppliers in the different Ethereum layers. There are however clear similarities between the number of suppliers per unique dependencies between clients written in the same ecosystem. For the Go ecosystem each supplier provides on average 1.7 and 2.0 unique artifacts for Geth and Prysm respectively. For the Rust ecosystem each supplier provides on average 1.5 and 1.3 unique artifacts for OpenEthereum and Lighthouse respectively. For the Java ecosystem this value is 2.7 and 2.4 for Besu and Teku respectively.

RQ2. What is the diversity of the software supply chain of Ethereum across ecosystems?

Looking at figures 2, 3, and 4, there is a drastic difference in software diversity between the studied ecosystems. For Go, shown in figure 2, there is nearly a complete overlap of unique dependencies. 95% of the unique dependencies which are required by Geth are also dependencies of the Prysm client. Of the overlapping dependencies the largest providers of monoculture are btcsuite, influxdata, mattn, and prometheus. Btcsuite provides a collection of tools for Bitcoin and cryptography in Go. Influxdata are an enterprise grade software provider which specializes in platforms for time series databases. Prometheus is also a provider of monitoring and time series database tools. Mattn is the largest provider of monoculture who is a sole developer.

The supply chains of the Java Ethereum clients are more diverse, as seen in figure 4, although the majority of unique dependencies are shared amongst both Teku and Besu. 69% of the unique dependencies in Teku, and 60% of the unique dependencies in Besu, are overlapping. The largest monoculture providers in the Java supply chain are Netty, OpenTelemetry, and Apache. Netty is by far the largest supplier for the Java clients, providing 30 dependencies which are required by both Teku and Besu. Netty provides APIs and tools for developing asynchronous server communication. OpenTelemetry is

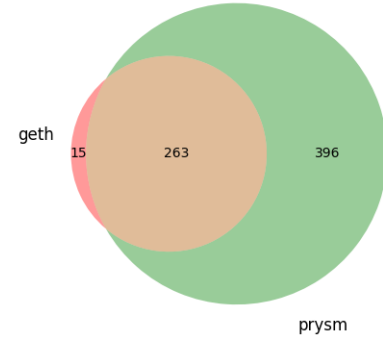


Figure 2. Intersection of Go Ethereum Dependencies

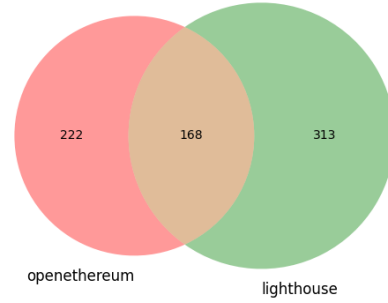


Figure 3. Intersection of Rust Ethereum Dependencies

a provider of APIs for monitoring and logging data. They are the second largest supplier of dependencies, providing 12 dependencies for both clients. Apache is a vast organization and one of the most prominent contributors of open source software and support a large number of various projects. Notable packages which Apache provides both Teku and Besu are Tuweni, which aids development of cryptography functions, and log4j, a utility for creating customized log messages for running processes.

The Rust Ethereum clients have the most diverse supply chain. The majority of unique dependencies are not in the intersect of OpenEthereum and Lighthouse. Only 43% of OpenEthereum dependencies and 35% of Lighthouse dependencies are overlapping. There are very few providers of monoculture in the Rust developed clients. The most prevalent providers in Rust are Crossbeam, Parity, and Serde. Crossbeam is a provider of tools for concurrent programming in Rust. Parity are providers of numerous tools used in blockchain development in Rust. Serde is a set of tools used for serializing and deserializing data structures, which is used for storing data or transferring data over networks.

VI. DISCUSSION

There is an apparent difference in size of the software supply chains of the Eth1 Execution layer and Eth2 Consensus layer of the Ethereum Blockchain. Although Eth1 was released 5 years before Eth2, and studies have shown that a project's dependencies tend to grow over time, the dependency metrics of the Eth2 clients are all larger than the Eth1 clients from the same ecosystem, save for Besu. This points to that the Eth2

Table III
SOFTWARE SUPPLY CHAIN METRICS OF ETHEREUM CLIENTS

Client	Programming Language	Total Dependencies	Unique Direct Dependencies	Unique Transitive Dependencies	Suppliers
Geth (Eth1)	Go	362	67	211	166
OpenEthereum (Eth1)	Rust	1447	67	382	299
Besu (Eth1)	Java	1473	63	149	80
Prysm (Eth2)	Go	901	123	536	328
Lighthouse (Eth2)	Rust	2044	77	435	387
Teku (Eth2)	Java	2998	59	178	99

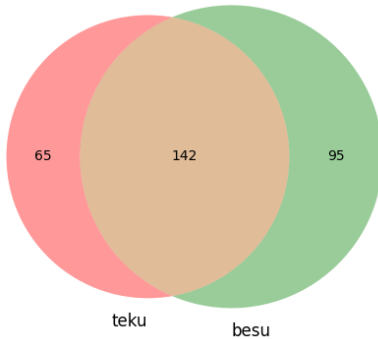


Figure 4. Intersection of Java Ethereum Dependencies

Consensus layer is a more complex system of cryptography procedures, requiring far more external software packages in order to function. This idea is supported best by the findings shown in figure 2, which shows that 95% of the supply chain of Geth only constitutes 40% of the supply chain of Prysm.

The gathered data also shows interesting trends between different ecosystems. As the clients within each Ethereum layer are functionally equal on a macro level, the study of their supply chains should reflect well on the ecosystems which they are built upon. From this study we can see that Java, the oldest of the studied ecosystems, is dominated by larger organisations who supply large amounts of Open Source software. This is assumed to be due to the effects of *hype driven development* over a long period of time. As time progresses, developers tend to choose software packages supplied by reputable vendors, and larger reputable organisations outlast and take over development from smaller vendors. Although Rust and Go are only released a year apart, 2010 and 2009 respectively, Rust is much more diverse in both software packages and suppliers. It is assumed that this is due to Go being maintained by a Google, a large corporation with more stringent software requirements, compared to Rust which is community driven.

It is self admitted that the Ethereum Foundations ambitions to achieve client diversity is far off the mark, however the data gathered in this paper points to that the software diversity is in a far worse state.

VII. CONCLUSION

In this paper, the first systematic analysis of the software diversity, with a focus on open-source software dependencies, in the Ethereum ecosystem is presented. The dependency trees of three pairs of Ethereum clients, developed in the languages Go, Rust, and Java, were collected and transformed

into a unified format. In this unified format, the dependency trees of the clients were analysed individually as well as together with clients developed in the same language. This analysis resulted in a novel data set of quantitative metrics describing this size of the Ethereum software supply chain. The data set shows that the Eth2 consensus layer requires a far greater amount of dependencies to function compared to the Eth1 execution layer. The data set is also used to show that there is a significant overlap of dependencies used by clients developed in the same language. This overlap was largest in the Go developed clients, where 95% of dependencies of the Eth1 client Geth were also dependencies of the Eth2 client Prysm. The smallest overlap seen was between the Rust clients OpenEthereum and Lighthouse, which shared 43% and 35% of their dependencies respectively.

ACKNOWLEDGEMENT

The author would like to thank the supervisors of this project, Benoit Baudry and César Soto-Valero, for their unwavering support, guidance, and words of encouragement throughout this project.

REFERENCES

- [1] C. Lamb and S. Zacchiroli, "Reproducible builds: Increasing the integrity of software supply chains," *IEEE Software*, pp. 1–10, 2021.
- [2] "2019 state of the software supply chain," Sonatype, Tech. Rep., 2019. [Online]. Available: https://www.sonatype.com/hubfs/SSC/2019%20SSC/SON_SSSC-Report-2019_jun16-DRAFT.pdf
- [3] W. Enck and L. Williams, "Top five challenges in software supply chain security: Observations from 30 industry and government organizations," *IEEE Security Privacy*, vol. 20, no. 2, pp. 96–100, 2022.
- [4] M. Ohm, H. Plate, A. Sykosch, and M. Meier, "Backstabber's knife collection: A review of open source software supply chain attacks," in *Detection of Intrusions and Malware, and Vulnerability Assessment*, C. Maurice, L. Bilge, G. Stringhini, and N. Neves, Eds. Cham: Springer International Publishing, 2020, pp. 23–43.
- [5] J. Yang, Y. Lee, and A. P. McDonald, *SolarWinds Software Supply Chain Security: Better Protection with Enforced Policies and Technologies*. Cham: Springer International Publishing, 2022, pp. 43–58. [Online]. Available: https://doi.org/10.1007/978-3-030-92317-4_4
- [6] A. Decan, T. Mens, and P. Grosjean, "An empirical comparison of dependency network evolution in seven software packaging ecosystems," *Empirical Software Engineering*, vol. 24, Feb 2019.
- [7] ENISA. (2021, Jul) Understanding the increase in supply chain security attacks. [Online]. Available: <https://www.enisa.europa.eu/news/enisa-news/understanding-the-increase-in-supply-chain-security-attacks>
- [8] F. B. Cohen, "Operating system protection through program evolution," *Comput. Secur.*, vol. 12, pp. 565–584, 1993.
- [9] S. Forrest, A. Somayaji, and D. Ackley, "Building diverse computer systems," 06 1997, pp. 67–72.
- [10] B. Baudry and M. Monperrus, "The multiple facets of software diversity: Recent developments in year 2000 and beyond," *ACM Comput. Surv.*, vol. 48, no. 1, Sep 2015. [Online]. Available: <https://doi.org/10.1145/2807593>

- [11] Open Source Initiative. (2007, Apr) The open source definition. [Online]. Available: <https://opensource.org/osd>
- [12] Ethereum Foundation. (2022, Feb) What is ethereum? [Online]. Available: <https://ethereum.org/en/what-is-ethereum/>
- [13] Ethereum Foundation. (2022, Feb) Client diversity. [Online]. Available: <https://ethereum.org/en/developers/docs/nodes-and-clients/client-diversity/>
- [14] A. Benelallam, N. Harrand, C. Soto-Valero, B. Baudry, and O. Barais, “The maven dependency graph: A temporal graph-based representation of maven central,” in *Proceedings of the 16th International Conference on Mining Software Repositories*, ser. MSR ’19. IEEE Press, 2019, p. 344–348. [Online]. Available: <https://doi.org/10.1109/MSR.2019.00060>
- [15] S. Benthall, T. Pinney, J. Herz, and K. Plummer, “An ecological approach to software supply chain risk management,” 01 2016, pp. 130–136.
- [16] D.-L. Vu, I. Pashchenko, F. Massacci, H. Plate, and A. Sabetta, “Typosquatting and combosquatting attacks on the python ecosystem,” in *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*, 2020, pp. 509–514.
- [17] C. Soto-Valero, T. Durieux, and B. Baudry, “A longitudinal analysis of bloated java dependencies,” in *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2021. New York, NY, USA: Association for Computing Machinery, 2021, p. 1021–1031. [Online]. Available: <https://doi.org/10.1145/3468264.3468589>
- [18] I. Pashchenko, H. Plate, S. E. Ponta, A. Sabetta, and F. Massacci, “Vulnerable open source dependencies: Counting those that matter,” in *Proceedings of the 12th International Symposium on Empirical Software Engineering and Measurement (ESEM)*, Oct 2018.
- [19] E. Zamani, Y. He, and M. Phillips, “On the security risks of the blockchain,” *Journal of Computer Information Systems*, vol. 60, no. 6, pp. 495–506, 2020. [Online]. Available: <https://doi.org/10.1080/08874417.2018.1538709>
- [20] J.-P. Aumasson, D. Kolegov, and E. Stathopoulou, “Security review of ethereum beacon clients,” *arXiv preprint arXiv:2109.11677*, 2021.
- [21] C. Soto-Valero, M. Monperrus, and B. Baudry, “The multi-billion dollar software supply chain of ethereum,” 2022. [Online]. Available: <https://arxiv.org/abs/2202.07029>
- [22] Mozilla. (2022, Apr) Working with json. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON>

Content 2022

Automated car following and platooning

Autonomous Robotic Systems

Learning in Dynamical Systems

Embedded Systems and Motor Drives for Electric Transportation

Semiconductors for Embedded Systems

Power System Control

Power System Planning and Electricity Markets

Design and Testing of Novel Microwave/Antenna technologies

Fusion – the Sun's Energy Source on Earth

Observations in Space Physics

Observation Platforms and Instrumentation for Space Physics

Artificial Intelligence for the Internet of Things

Information Engineering: Big Data & AI

Artificial Intelligence

Big Graphs of Software Packages