Overseeing Intersection System for Autonomous Vehicle Guidance

Alexander Adolfsson and Daniel Arrhénius

Abstract—Intersections represent one of the most common accident sites in traffic today. The biggest cause of accidents is obstructed view and subpar communication between vehicles. Since autonomous vehicles rely on sensors that require a direct view intersections are some of the most complex situations. Where the potential for inter vehicular communication exists between modern vehicles, it is absent in the older generation. An overseeing intersection system can fill this function during the transition period to fully autonomous traffic. This project aimed to implement an intersection system to assist autonomous vehicles through a crossroad. The assist system’s objective was to collect and transmit data from cars close to the junction to the autonomous vehicles nearby. The concept was tested in simulations by having models traverse a crossroad to evaluate how it utilised the external information. No persistent conclusion could be made due to insufficient simulation environment and vehicle model control.

Index Terms—Autonomous Vehicle, Intersection System, Road Safety, Self-Driving Cars.

TRITA number: TRITA-EECS-EX-2019:120

I. INTRODUCTION

In 2018, more than half of all fatal and injury crashes occurred at or near intersections in the US [1]. Increasing intersection safety can therefore lead to a large reduction in road fatalities and injuries. There are many ways to improve safety, from redesigning the intersections to applying technology and more. The introduction of 5G and the focus on smart cities will shift the transportation towards more automation, and therefore the amount of technology in intersections will likely increase as the vehicle can take more advantage of the information. This is especially true for major cities, where it’s hard to redesign old intersection but easy to install additionally technology to existing ones. Improvements in technology and legislation has already reduced the amount of deaths in junctions the last decade in Europa by 40%, from 8'463 to 5'142 [2], but more should be done. The most common reasons for accidents in intersections are information failure either among drivers, the driver and the traffic environment, or because obstructed view [3].

The next big leap in road safety will be dependent on better information sharing systems and vehicles that can act on this information to assist, or completely take over, the driving.

As autonomous vehicles get closer to the market there will be an overlap where autonomous and human driven vehicles will share the same road network. Many previous studies have been carried out with different designs. Most studies and current in-use technology focus on the cars own ability to read data from its surroundings. These autonomous vehicles use LIDARS, cameras and in some cases even ultrasonic sensors as well as radar. The problem with these technologies is that they require a, more or less, unobstructed view to give any useful data. So, when clear view is unobtainable another solution is required.

The main goal of this project is to build a simulation software with the support system to assist autonomous vehicles through a crossroad junction with multiple other vehicles of different types. The design thought is to simulate an overseeing intersection computer that receives, registers, and distribute information about all the vehicles in proximity to the intersection. The decision making is left to the cars own system, making the standardisation among different car manufacturers easier. This method could ease the initial introduction of autonomous vehicles in traffic without extra outfitting required on the older generation of vehicles.

In the following chapter the model and the underlying theory will be discussed. In Chapter II the theory will be discussed, detailing the control theory as well as the mathematics behind the path making. In Chapter III the model is the focus with an explanation of the Robotic Operating System, the basic bicycle model, the intersection system, and the general design of the detection and trajectory system. Chapter IV showcases some of the model’s capabilities and deficiencies. The following chapter, Chapter V, brings up the project to discussion, talking about the model, alternative way of solving the problem, and future work. In Chapter VI the project members draw a conclusion of the project.

II. THEORY

A. Proportional Control

A proportional controller, also called p-controller, is a feedback control system where the input to the system is dependent on the output, see Fig. 1. For a p-controller the feedback is used to calculate the error between the current and the desired output.

\[ u(t) = K_p e(t) + u_0 \]  

(1)

where \( u_0 \) is the controller output with zero error, \( e(t) = SP - PV \) is the difference between the set point, \( SP \), and

\[ PV \]

Fig. 1. Block diagram of a P-controller.

The equation for a p-controller is the following
the process variable \( PV \), and \( K_p \) is the proportional gain constant [4]. The advantages of a p-controller are that it is easy to compensate for the disturbance in a system without using advanced control theory. The disadvantage is that the error may not be eliminated with only a p-controller. This can be fixed with more advanced controllers such as PD-controller or PID-controller. The PID-controller can to some extent remove the error and lower the rise time, making the system faster, but is more complex to implement. However, if a simpler controller works with a small enough error and rise time then it is not desired for a more advanced controller.

B. Curve Fitting

Curve fitting is a method to match an equation to coordinates and is often used to predict patterns in collected data. There are many different types of curves that are possible to get using curve fit depending on the equation chosen. For curved paths a quadratic curve fit is the most common and the equation is the following

\[
x_2 = ax_1^2 + bx_1 + c
\]

where \( x_1, x_2 \) denotes coordinates and \( a, b, c \) are constants that the curve fitting method calculates for the optimal match.

III. Model

The vehicle’s parameters are based on the small-scale vehicles at KTH Smart Mobility Lab, SML for short, see Table I. The model constitutes the scale of the relative parameters for the simulation.

A. Robot Operating System

Robot Operating System, known as ROS, is an open source operating system commonly used in robotic experiments. It contains a collection of tools and libraries that are meant to simplify the process of designing a complex and robust robot software. The basic software structure that this project is based on is from the SML and is meant to enable the same code used for simulation to run on the small-scale vehicles with only minor modifications. This enables one to check if the software works as intended before trying on the actual vehicles, minimising the risk of errors in the code. ROS can be used with the programming languages C++ or Python 2.7 [5].

B. The Kinematic Bicycle Model

The kinematic bicycle model is a simplified model of a car assuming only two wheels instead of the more advanced car models. The advantages of using a simplified model is the ease of usage for simulation and practical applications. The model lumps the front and rear wheels together into one front and one rear wheel located on the centre, see Fig. 2. Acceleration \( u_1 \) and front wheel steering angle \( u_2 \), in Fig. 2 \( u_2 = \delta_f \), are the only control inputs. The model can be written [6] as

\[
\begin{align*}
\dot{X} &= V \cos (\psi + \beta(u_2)) \\
\dot{Y} &= V \sin (\psi + \beta(u_2)) \\
\dot{V} &= u_1 \\
\dot{\psi} &= \frac{V}{l_r} \sin (\beta(u_2)) \\
\end{align*}
\]

where \( \beta(u_2) \) is the slip angle at the centre of gravity

\[
\beta(u_2) = \arctan \left( \frac{\tan(u_2) l_r}{l_f + l_r} \right).
\]

![Fig. 2. The kinematic bicycle model of the vehicle [6]](image)

For lessening the errors obtained with this model compared to more accurate vehicle models the acceleration was limited to less than 0.5g in accordance to [6]. The constants in Fig. 2 is found in Table I where G is centre of mass, \( l_r \) is the distance from the centre of mass to the rear wheel, \( l_f \) is the distance from the centre of the front wheel to the centre of mass, and \( |\delta_f| \) is max steering angle.

C. Intersection System

The intersection system’s purpose is to collect and relay information from traffic towards any vehicle capable of receiving the data. The desired information is velocity, position, direction, and if possible, destination of each vehicle. This can be achieved by having the vehicles capable of producing its own data sending it to the intersection or using a system for the older vehicles, such as an infrared system, capable of tracking the traffic inside the intersection to cover potential blind zones for the autonomous vehicles’ sensors. The vehicle to intersection communication can be a similar concept to the vehicle to vehicle, in short V2V, communication researched by [7]. The specific way in which the intersection acquires the vehicle information is not important for the simulation and different systems will be brought up for discussion in Section V-B.
TABLE II
CONCERNED DISTANCES AND ACTION FOR THE DIFFERENT ZONES, SEE FIG. 3

<table>
<thead>
<tr>
<th>Warning Zone</th>
<th>Distance [m]</th>
<th>Direction</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zone 1</td>
<td>&lt; 0.7</td>
<td>Any</td>
<td>Stop</td>
</tr>
<tr>
<td></td>
<td>0.7 &lt; l &lt; 2</td>
<td>Towards Vehicle</td>
<td>Slow</td>
</tr>
<tr>
<td></td>
<td>0.7 &lt; l &lt; 2</td>
<td>Same as Vehicle</td>
<td>Dependent on Velocity, See Fig. 4</td>
</tr>
<tr>
<td>Zone 2 &amp; 6</td>
<td>&lt; 0.7</td>
<td>Any</td>
<td>Stop</td>
</tr>
<tr>
<td></td>
<td>0.7 &lt; l &lt; 2</td>
<td>Towards Vehicle</td>
<td>Track</td>
</tr>
<tr>
<td>Zone 3 &amp; 5</td>
<td>&lt; 0.7</td>
<td>Any</td>
<td>Stop</td>
</tr>
<tr>
<td></td>
<td>0.7 &lt; l &lt; 2</td>
<td>Towards Vehicle</td>
<td>Track</td>
</tr>
<tr>
<td>Zone 4</td>
<td>&lt; 0.7</td>
<td>Any</td>
<td>Stop</td>
</tr>
<tr>
<td></td>
<td>0.7 &lt; l &lt; 2</td>
<td>Towards Vehicle</td>
<td>Track</td>
</tr>
</tbody>
</table>

D. Simulated Sensors

The program creates and uses data with the assumption that it represents the sensory output data from an external system. This data is used complementary to the data feed from the intersection system described above. The created sensory data includes Light detection and ranging, LIDAR, as well as GPS. The manufactured LIDAR data is used by the vehicles close proximity detection system, using the distance and the angle relative from the car to determine actions. More details about the vehicle detection systems in Section III-E. The GPS data is used to find and create a path for the vehicle.

E. Vehicle Detection System

The vehicle detection system is divided into different warning zones that corresponds to relative angle between vehicle and danger as [8] did in their study, see Fig. 3. The forward relative angular is \( \theta_f = 30^\circ \) and the rear relative angle is \( \theta_r = 30^\circ \). Dividing the area around the vehicle into different zones help the vehicle to know where to focus its attention.

A target has different dangers towards the vehicle dependent on its distance, relative angle and directions. The vehicle classifies each target it detects with a specific threat level. The higher the threat level a target has the more attention it’s given by the vehicle. The threat level ranges from 0, meaning no risk of collision in the approximate future and no action taken, to 5, meaning the object is within a radius of 0.35m of the centre of mass and a emergency break action is instigated to prevent collision.

The different actions the vehicle can perform depends on the circumstances in which it finds itself in. The different distances and actions taken can be seen in Table II. The “Track” action in the table implies that only during certain action or manoeuvres is the target considered a threat. Fig. 4 shows how the vehicle detection system works in the forward warning zone.

F. Intersection

The intersection is made from placed points of interest such as start and stop points for the lanes and roads. The intersection is made to have all the parameters flexible, the number of lanes, connecting road length and width, and traffic rules. These points of interest are later used by the car’s pathfinder system to create a general course. For the experiments the default setup used was a four-way crossing with three lanes and with a road length of 10m and width of 7m. A graphic of the intersection can be seen in Fig. 5. This in relation to the SML car would be a road length of about 17 cars and lane width of 1.67 cars. Given that an average car has a width of 1.84 meters [9] and the minimum lane width is 3.0m, each lane corresponds to 1.63 cars.

G. Pathfinder

The pathfinder has two phases. The first part makes a general plan of the vehicle’s route. This path is divided into smaller segments between general points of interest in the intersection. The second phase creates a more detailed path from the vehicle’s current position to the end of the current section of the path. The pathfinder is designed to represent the GPS giving coordinates to the vehicle, which then makes a more thorough plan of direction. The detailed path is made by taking making a curve fit, as detailed in Section II-B, between the vehicle’s position and the closest point of interest. The fitted curve is later fed into a controller that generates a change of angle and velocity to follow the path. The pathfinder is also responsible for changing the path when obstacles get in the way, updating it with new points of interest to avoid collision.

H. Vehicle Types

The program has two different types of vehicle behaviours. The first being autonomous, which uses the sensory input, II2V data, and tracking logic presented above to traverse the intersection without colliding with the other vehicles. The second type is manually driven, which is given a path and instructed to follow it, without regard of its surroundings.

IV. SIMULATIONS AND RESULTS

Results from simulations of different scenarios will be presented in this section. First a benchmark is performed, to set a time reference for the other trials. The succeeding test are designed to maximise interaction between the vehicles in realistic scenarios. The main objective is to reduce the risk of collision and doing so without losing efficiency. The distance from the main car to the closest obstacle is used as a measurement of safety, and the time required to finish the designated route is used as a measurement of efficiency. In the simulations the main vehicle always starts in the same position, the first outer lane of the left road, with the same instructions while the obstacle car is moved in between simulations to create new scenarios. The distance measurements are taken from the internal tracking system used by the autonomous vehicle. The distance is set from the car’s centre, rather than the hull. In the simulations presented the main car is an autonomous vehicle type and the obstacle vehicle is of the manually driven type.

A. Benchmark and Time Samples

Having the model do the requested path without losing unwarranted efficiency is important. As external data from other studies or observations is hard to use when deeming the
efficiency of the model and, more important, the subsystems responsible to keep it from causing accidents basic sampling was performed and used as a benchmark. Table III shows the mean time expenditure of the different scenarios with iterations. The obstacle vehicle’s velocity is set to 60% of main vehicle’s target speed to have it below the trigger point for the overtaking action, making the time expended more meaningful for efficiency since the vehicle is not deliberately slowed down.

### TABLE III

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Mean [s]</th>
<th>Median [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benchmark Left Turn</td>
<td>0.879</td>
<td>0.85</td>
</tr>
<tr>
<td>Overtake</td>
<td>0.909</td>
<td>0.91</td>
</tr>
<tr>
<td>Obstacle from right collision course</td>
<td>0.894</td>
<td>0.89</td>
</tr>
<tr>
<td>Obstacle from in front collision course</td>
<td>0.912</td>
<td>0.91</td>
</tr>
<tr>
<td>Obstacle from left collision course</td>
<td>1.175</td>
<td>1.14</td>
</tr>
</tbody>
</table>

The first scenario shows a secondary vehicle at low speed with the same initial path as the main vehicle. The scenario after the overtaking is instigated is shown in Fig. 6. Fig. 7 shows the distance between the cars during the overtake and the proceeding route.
C. Obstacle from Right Collision Course

The second scenario is a secondary car, coming from the road to the right of the main car, set on a collision course in the middle of the intersection. The scenario is shown in Fig. 8 and sample of distance shown in Fig. 9. This scenario is problematic since the secondary car’s path intersects the main car’s even after a lane change. This means that the car gets closer and must initiate emergency breaks, and for some distances this will not be enough as the secondary car will hit from the side. In a real traffic situation, the main car would not be held accountable for the accident, but it’s still not the ideal results of a simulation.

D. Obstacle from in Front Collision Course

The third scenario is a secondary car, coming from across the junction and crossing the path of the main car. The scenario and sample distance can be seen in Fig. 10 and Fig. 11, respectively. In this setup there is an approximate 10% of the initial position that can result in a collision. This happens when the simulation is in such a configuration that the secondary car ends up in a situation where it triggers change lane subroutine and occasionally collides.

E. Obstacle from Left Collision Course

In the last scenario there is a secondary car from the left doing a left turn through the intersection, see Fig. 12. The main car has nowhere to go so is left to slow down and break to a complete stop to avoid collision. In some cases, the simulation ends up in a position that the cars go from safe to collision within two frames. This makes it impossible for the code to
break or adjust trajectory. This happens because the bicycle model appears acts independent of the rest of the program. The distance from such an example is show in Fig. 13.

![Fig. 12. Mid-simulation of autonomous car from the left on collision course with left heading car from left](image)

![Fig. 13. Distance between main and secondary car from a collision when secondary vehicle comes from left](image)

**V. DISCUSSION**

**A. Model**

The results show that the system can control a vehicle in difficult situations, but since there is no baseline to compare it to it’s hard to tell the efficiency of the model. The only value that could be used for comparison is from basic mathematical models, estimations, and trial runs in the simulation. To get values that relates to reality a more realistic model would be required.

**B. Different Intersection Systems**

The intersection system is the main part of this project and there are several different ways to implement it considering current technology and pricing. The cheapest system possible to implement is a vehicle to intersection, V2I in short, communication system, where the car sends information to the system that then distribute the information to other vehicles. The only difference with this system compared to the V2V system tested by [7] is the range and no obstacle, such as buildings, blocking the signal. The biggest disadvantage with this type of system, for both V2V and V2I, is that it is dependent on vehicles capable of communicating and receiving information from outside sources. Because of this disadvantage, the V2I system will probably never be a sole system for larger intersection and for smaller intersection the V2V system, that is necessary for the V2I system, will be enough. It can however be part of a bigger system.

A different system is the active infrared sensor, called IR sensor, system, which is currently used in certain intersections, together with a communication system like the V2I system. The advantage of an active IR sensor is its ability to measure distance and velocity of target vehicle during both day and night while not costing as much as most system [10]. The pricing information in [10] is a few years old and technologies have become cheaper, especially camera system. Still, with modern pricing the IR sensors are cheaper than most alternatives. Another advantage with a system of IR sensors is its ability to cover both a large intersection, with several lanes, as well as a small intersections without modifications to the setup. The largest disadvantage of the IR system is the inability to survey the intersection in case of incidents. The IR sensor system together with a communication system capable of sending information to incoming vehicle would be a possible setup.

The technology that have had the highest advancement in recent time is the usage of camera technology together with advanced software, such as machine learning, to identify and track objects. Setting up a video camera facing each direction of the intersection could track and calculate far more than an IR sensor system for example as more information is gathered. The camera system also has the advantage of surveying the intersection. With the usage of several more video cameras the tracking of pedestrians and bicyclist can also be achieved to reduce accidents. A survey of pedestrian detection processes by [11] show several methods with high accuracy, over 90%, in 2014 making it a possible implementation in modern surveillance system. The largest disadvantage of such a system is the cost of the equipment, see [10], and the advanced software must be able to differentiate between all vehicles, bicycles, and pedestrians while also calculating vehicle distance and velocity. A lesser problem is the difficulty for video cameras to operate in darkness. This can easily be solved by placing streetlight, or another light source, in the vicinity of the intersection. Another problem is the privacy of personal information with technology capable of identifying and tracking persons. This privacy question is very new and have not been studied extensively, the best region to study is China where an extensive surveillance system exists. One survey on residents in Beijing, China by [12] shows that people of different age and gender have different opinions on surveillance but states that more research is necessary in the subject. The video camera system is the most advanced and will likely be implemented in important intersections, such as in the city where traffic is complex, together with a communication system for the same reasons the IR system needed one.
C. Expanding the Project

The model can perform manoeuvres to deal with certain situation, but far from all of them. There are still some complex scenarios the vehicle can’t manage without engaging the emergency breaks and/or being hit. In further development of the model some minor control of the secondary cars should be implemented, making them more reactive to the dynamic model of the main car. Beyond that additional work could be put into the path planning, making it more foreseeing and dependent on the path of the other cars when that information is available. It could also be reason to use the model and implement some minor slot-based planning by the central computer. That would ensure efficiency for all the autonomous cars, but some well-planned design would be required to make sure the manual cars would still have place in the finished product.

VI. CONCLUSION

The model has potential to improve road safety, even though no concussive results were produced by this study. The problem with the model is not that the concept is lacking, but that the vehicle control is insufficient. It is recommended that further studies be conducted in the area to find the most efficient model, both in safety and in monetary cost. The created model could be improved upon and used for further tests, with improved subroutines for the vehicles.

ACKNOWLEDGMENT

The authors would like to thank our supervisor Joana Filipa Gouveia Fonseca for her guidance during this project. Her help with control theory and project planning made this project possible. Another great source of guidance was from Frank Jiang whom showed us how ROS and the SML worked.

REFERENCES