Linköping University | Department of Computer and Information Science
Master's thesis, 30 ECTS | Computer Science and Information Technology
22 | LIU-IDA/LITH-EX-A--22/031--SE

Empirical Data Based Predictive Warning System on an Automated Guided Vehicle

Empiriskt databaserat predikterande varningssystem för självkörande truckar

Anton Blåberg
Gustav Lindahl

Supervisor : Sofia Thunberg
Examiner : Tom Ziemke

External supervisor : Michal Godymirski, Toyota Material Handling
External supervisor : Amanda Nilsson, Toyota Material Handling

LINKÖPING UNIVERSITY

Linköpings universitet
SE–581 83 Linköping
+46 13 28 10 00 , www.liu.se

Abstract

An Automated Guided Vehicle (AGV) must follow protective regulations to avoid crashing into people when autonomously driving in industries. These safety norms require AGVs to enable protective fields, which perform hard braking when objects enter a specific area in front of the vehicle. Warning fields, or warning systems, are similar fields that decrease the speed of the AGV before objects enter the protective fields to enable a steadier driving. Today at Toyota Material Handling Manufacturing Sweden (TMHMS), warning systems have been implemented but the systems are too sensitive to objects outside of the AGVs path.

The purpose of this thesis is to develop a predictive warning system based on empirical data from previous driving scenarios. By storing previous positions, the warning system could estimate a trajectory based on simple statistics and deploy speed limiting decisions if objects appear in the upcoming predicted path.

The predictive warning system was compared to the current warning system and a deactivated warning system setup in driving performance and driving dynamics. Performance was measured by measuring time to finish an industry-like test track and dynamics was subjectively rated from a group of experienced AGV developers from TMHMS. Results showed that a predictive warning system drove the test track faster and with better dynamics than the current warning system and the no warning system setup.

Key findings are that a predictive warning system based on empirical data performed better in most cases but has some extra requirements to function. Firstly, the method require the AGV to mostly drive on previously driven paths to produce good results. Secondly the warning system requires a somewhat powerful on board computer to handle the computations. Finally, the warning system requires spatial awareness of pose for the vehicle, as well as structure and shape for deployed protective fields.

Acknowledgments

This thesis has been supported by a large group of people. Firstly, we are very grateful to our examinator **Tom Ziemke** for making sure we got off to a good start. You gave us great ideas in evaluating our system as well. A very big thanks goes to our LiU supervisor **Sofia Thunberg** who have played a vital part in understanding literature, thesis structure and thesis grammar. We were both very impressed at the speed you were able to help us at.

Secondly, we are very grateful to all the people at Toyota Material Handling Manufacturing Sweden who have asked great questions and helped us in so many ways. A special thank you goes out to our supervisors **Amanda Nilsson** and **Michal Godymirski** who have helped us with everything from answering every little question to helping us mapping out our method by the whiteboard. They have been there every day and without them the quality of this thesis would be far worse. Special thanks to **Filip Sundqvist** for helping us tame the machines.

We would also like to say thank you to **Ester Brandås**, **Sandra Ljungberg**, **Erik Sellén** and **Robert Sehlstedt** who were also thesis students at Toyota for keeping us company at our lunch breaks and for fikas when everything else was tough.

Furthermore, I (Anton) would like to thank my nearest friends and family with a special thanks to my wife **Alice Blåberg** who always cheered me on.

Finally, I (Gustav) would also like to say a personally thank you to my family and closest friends but especially my fiancé **Ewelina Bladh** who has been there as a pillar every day when I have gotten home.

Contents

List of Figures

List of Tables

Abbreviations

AGV	Automated Guided Vehicle
CDI	Carrier Drone Ion
CWS	Current Warning System
FIS	Fuzzy Inference System
FPFP	Future Protective Field Prediction
LiDAR	Light Detection and Ranging
NWS	No Warning System
PWS	Predictive Warning System
TMHMS	Toyota Material Handling Manufacturing Sweden
TMS	Theoretical Maximum Speed
TP	Trajectory Prediction

1 Introduction

The introduction will first present the motivation behind the thesis followed by the aim, and research questions. Finally the delimitations and thesis outline are presented.

1.1 Motivation

In many industry settings in recent years, there has been transition from manual labor to labor from automated machines. There are many benefits for machines to perform dangerous and repetitive tasks instead of a human. Machines make fewer mistakes, reduces human injury rates, and do not need coffee to run at a high pace. One example of such machines is Automated Guided Vehicles (AGVs), which are commonly used today in many industries. In short, AGVs are driverless vehicles programmed to perform different tasks and they are often used for transporting goods or materials. However, there are many issues with driving performance for these kinds of vehicles.

AGVs are often equipped with sensors to enable smooth and collision-free driving. One common sensor in AGVs is Light Detection and Ranging (LiDAR) sensors. The LiDAR sensor rapidly projects laser in its environment to find where the vehicle is positioned and measures distance to object in its surrounding. By doing so, the vehicles can have a warning system, such as different warning fields that can regulate the velocity of the vehicle depending on the data from the LiDAR sensor.

When objects enter the vehicle's warning field, the AGV will make decisions for an appropriate speed to drive in. The controller that decides the speed to drive in must be programmed by either the supplier of the scanner or the manufacturer of the vehicle. The industry trend is to have less and less available information on the vehicles themselves, resulting in less information to aid the controller in its decision making. This makes the overlaying system giving drive orders more modular and the AGVs more independent, since the system can be easier to change for different customers. The controller of the warning field affects the driving performance of the AGV, for example a too strict controller may produce "jittery" driving when objects shortly intervene the vehicles path. The reason for this behavior would be because of the length of the warning fields is often based on different speed intervals of the AGV, meaning there are a discrete amount of fields. Thus, if something appeared inside the field, the vehicle would slow down, and when it has slowed down the object would be outside of the field and the AGV would accelerate again. This unwanted behavior would go on in a repeating pattern until the AGV has passed the object.

One way to avoid this problem would be to predict what would happen if the velocity were increased. This would then be a predictive warning system. If the vehicle then looked further into the future, meaning where it will be in x number of seconds, it could also see what velocity it would need at that moment. For this to happen the AGV must know where it is heading, which can be difficult. This could be solved by predicting the future path of the AGV which is one thing this thesis investigates. That is, a new warning system that controls the highest allowed speed based on objects in the future path to improve driving performance of AGVs.

1.2 Aim

This thesis aims to explore a new method of controlling the velocity with a Predictive Warning System (PWS) to improve the driving performance and dynamics of an AGV at Toyota Material Handling Manufacturing Sweden (TMHMS). The developed warning system will be implemented in a real AGV and evaluated against the Current Warning System (CWS), as well as a setup with No Warning System (NWS). The evaluation will measure the AGVs time to complete a specific track, as well as letting an experienced jury subjectively grade the driving dynamics which were defined as perceived safety, smoothness and overall quality. Smoothness in this context is how well the AGV can drive without any jerkiness as well as a seamless change in velocities.

1.3 Research questions

The research questions for this thesis are the following:

1. How can a predictive warning system based on empirical data be implemented on a warehouse AGV at TMHMS?
2. How would such a warning system, compared to the current warning system at TMHMS, perform in the following aspects:
   a) *Driving performance:* Time to finish a short industry-like test track?
   b) *Driving dynamics:* Perceived safety, driving smoothness and overall quality?

1.4 Delimitations

The number of predictive warning systems methods that were implemented and evaluated were limited to one. Instead, three types of warning systems have been studied and used for evaluation. The first system was currently in use at TMHMS and the second Was be the new system presented in this thesis. The last evaluated system was a system with warning systems disabled. Another delimitation was that tests would only be performed on one track, as space is limited for AGV driving tests and rebuilding different tracks will be too cumbersome for this period. It is important to remember that this thesis investigates one form of AGV at TMHMS that uses a 2D LiDAR scanner. A final delimitation would be that even though the testing track tries to replicate the reality, it cannot recreate every case that could appear in a factory environment.

2 Background

This chapter will first describe the industry context for this thesis. Secondly, an AGV system overview is presented together with details of components on board the AGV. Finally, the concept of warning fields will be described.

2.1 Industry Context

This thesis has been carried out together with Toyota Material Handling Manufacturing Sweden AB (TMHMS) in Mjölby, Sweden. TMHMS is world leader in production of forklift trucks and produces a wide fleet of trucks, from small handheld trucks to rideable high reaching electric forklift trucks. TMHMS which belongs to Toyota Material Handling Europe (TMHE) which also has facilities in Italy and France, while operating under Toyota Industries Corporations. In the latest years Toyota have started developing their autonomous truck for logistics in warehouses. These are manual trucks with added sensors and software modules to handle the automation. Today, there is a possibility to turn some of TMHMSs electric forklifts autonomous and all TMHMSs automated trucks can also be driven manually when needed, for example if the truck is in the way of other trucks or people. Instead of looking at these trucks that can go in both modes, this thesis will focus on autonomous trucks that does not contain physical manual controls and were developed only to be autonomous. These trucks are not yet in production and therefore the name and implementation may differ in the future. Going forward in the thesis we will refer to our autonomous vehicle as Carrier Drone Ion (CDI) and details of this vehicle are presented in Section 2.2.

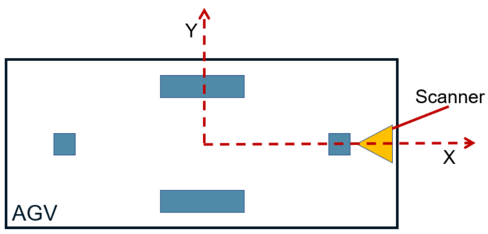
2.2 Details of the CDI

The CDI is a two wheeled differential driven AGV that is approximately half a meter tall and has a rectangular shape with side lengths of 0.8 and 1.4 meters as seen in Figure 2.1a, more details of the CDI can be found in Table 2.1. The purpose of this machine is to drive between locations in a factory, for example from a warehouse to a production line. An example of its work cycle would be that it picks up a pallet that another automatic forklift truck or person has placed on a loading station. It would then drive that pallet to a production line where another person or truck would take the pallet or the contents of the pallet. It would then repeat this process.

Table 2.1: Specifications for the CDI at TMHMS.

Features	Value	Unit
Width	0.812	[m]
Length	1.390	[m]
Height	0.530	[m]
Distance above ground	0.052	[m]
Number of steering wheels	2	[-]
Number of support wheels	2	[-]
Max velocity	1.680	[m/s]
Max carrying weight	1200	[kg]
Scanner offset from center to front	0.520	[m]
Scanner offset from center to side	0	[m]
Wheels offset from center to front	0	[m]
Wheels offset from center to side	± 0.345	[m]

(a) Photo of the CDI.



(b) Sketch of the CDI.

Figure 2.1: Images of the CDI, both real image and sketch.

2.3 System Overview

The CDI is built upon different programs running concurrently, each with different goals. For example, one program handles the loading and off-loading of pallets while another gets information about the path it will take. One of the more important modules is the communication software that keeps track of these programs and other information about the machine such as current speed of each wheel and information about the machine. The communication software is then capable of sending information to the different programs when they request the data via TCP and with an IP address, it can also receive data from these different programs. One of these programs that send information to the communication software is the positioning software which keeps track of the CDIs location and heading among other things. The on board warning systems tells the communication software to change its velocity depending on detected objects in its warning fields. This warning system is currently implemented in the on-board scanner hardware. An overview of how these systems is connected can be seen in Figure 2.2.

2.4 LiDAR and the SICK Microscan3

Light Detection and Ranging (LiDAR) is a technology used to find objects and the distance towards them. It works by sending different light beams and then detecting them when they reflect. The LiDAR scanner can then measure how long it takes for the beam to come back

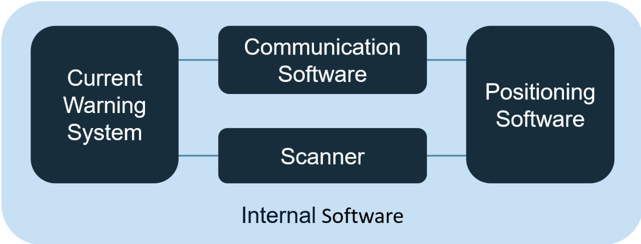


Figure 2.2: A system overview of the underlying software on the CDI.

and thus calculate the distance. If the light beam does not reflect the scanner will assume that no object exists in that direction. Common uses for LiDAR are on airplanes or helicopters and are used to survey the area. It can also have other application areas such as in smartphones or auto vehicles, such as AGVs [24].

The scanner used to generate warning field is called Sick Microscan3 and can be seen in Figure 2.3. This specific scanner that has been used in this thesis is the SICK MIC3-CBAZ40ZA1P01. This scanner is mounted at the front of the vehicle, see Figure 2.1b. The scanner scans the surroundings with the help of LiDAR with a minimum angle of $-47.5°$and a maximum angle of $227.5°$, with the angle relative to the x-axis from Figure 2.1b, with positive values going in counterclockwise rotation. For more details on the specifications of the scanner please see Table 2.2.

Figure 2.3: The SICK Microscan3 that is mounted on the CDI [18].

2.5 Warning and Protective Fields of Today

The size of the warning fields of today are discretized based on pairs of individual wheel speeds. These speeds are then placed into intervals. Example intervals for a left turn can be seen in Figure 2.4. As a differential drive vehicle turns by adjusting individual wheel speeds, a right turn is performed by driving the left wheel faster than the right wheel. For every combination of speed and turn rate, three different fields, shown in red, orange and yellow, are visible and will now be further explained.

The inner (red) field is the *protective field* and when objects enter the *protective field*, the CDI will initiate a hard braking and will always go to zero velocity. During this hard braking, the CDI observes if it is braking hard enough to keep the desired deceleration. This observation

Table 2.2: Specifications for the SICK MIC3-CBAZ40ZA1P01 [18].

Features	**Value**	**Unit**
Protective field range	4	[m]
Warning field range	40	[m]
Number of simultaneously monitored fields	$\leqslant 8$	[-]
Number of fields	128	[-]
Number of monitoring cases	128	[-]
Scanning angle	275	[deg]
Resolution	$30 - 200$	[mm]
Angular resolution	0.39	[deg]
Response time	$\geqslant 95$	[ms]
Protective field supplement	65	[mm]

will have one of two outcomes. Firstly, if the CDI observes its current deceleration is not sufficient enough to stop in time, it will instead initiate a harder braking by mechanically locking the wheels. When stopping the wheels in said way, the CDI has performed an *emergency stop*, which requires a manual restart to continue driving the CDI. Secondly, if the CDI has close to desired deceleration during the braking process the CDI will continue to brake until it is at zero speed. This way, the CDI has performed a protective stop, which does not require manual restart to continue driving the CDI.

The *protective fields* at TMHMS are generated by looking at where the CDI would be if it initiated a braking sequence down to zero speed at that current moment. Since the fields are separated into a small number of intervals based on speed the CDI groups many positional outcomes into one field and that is what can make them so wide in some cases. These fields are also calculated based on a worst-case scenario from the moment the scanner notices the object till it sends a signal to the CDI till it initiates braking and then finally stands still. According to ISO 3691-4:2020 [9] (which is a standard for industrial autonomous trucks) it is required have *protective fields* to be able to drive any AGV over 0.3 m/s. *Protective field* is the field that ensures personnel safety, and this field will not be altered in any way for this thesis and thus safety is not the focus of this thesis.

The middle (orange) field in Figure 2.4 is a so called *slow-down field*. When objects enter the *slow-down field* the CDI will, like the protective field, start to slow down. The difference is that the protective field will trigger an action to stop the CDI, whereas the *slow-down* field will only decelerate until the object has left the *slow-down field*. This field is generated by looking at what moment the CDI would have to start slowing down for the machine to change fields to avoid a *protective stop*.

The outer (yellow) field in Figure 2.4 is a so called *do-not-accelerate field*. When the scanner sees objects in this field the CDI may maintain its velocity or slow down, but it may not accelerate any further. This field is an extension to the *slow-down field* where the padding is equal to the distance the machine would travel to reach zero acceleration. This is done because one can then assume a constant velocity when the CDI reaches the *slow-down field*.

The *slow-down field* and *do-not-accelerate field* are what makes the warning fields of today. These fields' purposes are not to improve human safety or prevent crashes like the protective field. The purpose of warning fields is to have smooth driving dynamics and prevent hard braking and is discussed further in subsection 7.1.4. Warning fields are not required to be enabled to drive an AGV, and therefore can be changed to improve driving dynamics, hence the aim of this thesis.

TMHMS has observed weaknesses in both driving dynamics and performance in the CDI stemming from the warning fields and the weaknesses can be concatenated into three problems.

- Not keeping constant speed while driving close to a wall at low speeds.
- Slowing down because of a too wide warning field in turns close to walls.
- Slowing down for a wall when a turn is coming up.

These problems are the foundation for developing a new version of the warning fields. The first problem regarding constant speed can be explained with help from the leftmost column "Straight" in Figure 2.4. Imagine a CDI is driving slow in a straight narrow aisle between two pallet racks and the warning fields does not touch the racks. When the CDI increases in speed (see Straight-Fast cell in Figure 2.4) the warning fields will not only increase in length but also in width. By increasing the width of the warning fields, the CDI's field will now touch the pallet racks. When the warning fields touch the pallet racks, the CDI will slow down, which in turn will decrease the width of the warning fields. With more narrow warning fields the pallet racks are no longer inside the warning fields, resulting in the CDI trying to drive faster. In the scenario described above the CDI will never converge to a constant speed. This vicious cycle can be seen in Figure 2.5.

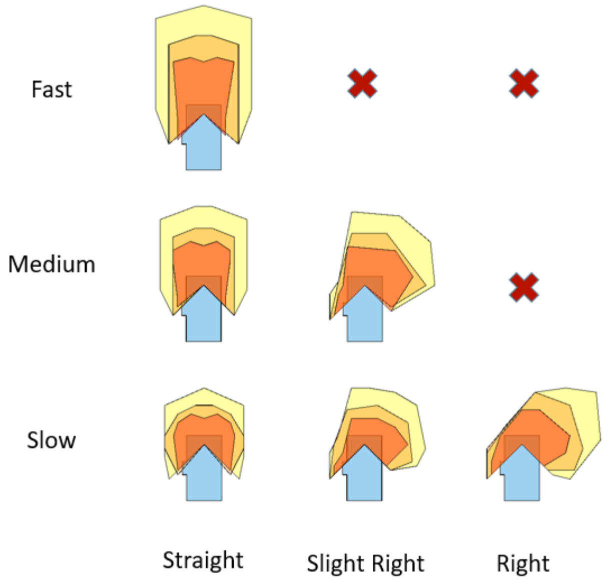


Figure 2.4: Discretized intervals for warning fields (yellow and orange) and protective field (red).

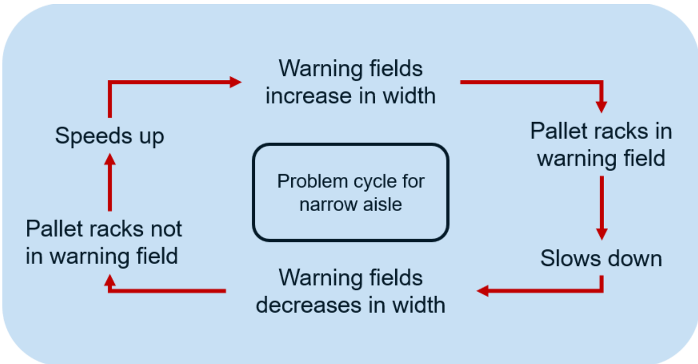


Figure 2.5: Decision problem when driving in narrow aisle preventing CDI from driving at constant speed.

3 Related Work

This chapter will present research related to our thesis. It will start with explaining concepts found in related research. After the concepts it will present how others handled trajectory prediction. Then studies in the topic of warning field generation will be presented. Finally, different solutions for decision making will be shown.

3.1 Concepts

This section explains concepts found in the literature.

3.1.1 Kalman Filters

A Kalman filter is a powerful method for estimating states in linear and nonlinear systems and has seen wide usages in the area of control theory [22]. Kalman filters has proven to be very effective at filtering out gaussian white noise [16]. Some systems enable measurement of all state variables but for some systems, state measurements cannot be performed. Where only input and output variables can be measured, a reconstruction of the system may be wanted. One method of reconstructing a system's states is estimation through an observer as seen in the following equations. Given the generic system model with system matrix A, control matrix B, output matrix C, system state x and system output y:

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx,\end{aligned}$$

it is possible to estimate the systems states with the following observer:

$$\dot{\hat{x}} = A\hat{x} + Bu + K(y - C\hat{x}).$$

With an approximation error of $\tilde{x} = x - \hat{x}$ the following differential equation is given:

$$\dot{\tilde{x}} = (A - KC)\tilde{x},$$

where K can be chosen for desired pole placement of $A - KC$ matrix. Let us now introduce a new system model:

$$\dot{x} = Ax + Bu + e$$
$$y = Cx + v,$$

where v is white noise measurement error which generates an interference e on the state vectors. To reconstruct state observer for this new system the approximation error now takes the following form:

$$\dot{\tilde{x}} = (A - KC)\tilde{x} + e + Kv,$$

with e and v having covariance matrix R_1 and R_2 respectively. The covariance matrix P for $\tilde{x}$ can now be calculated. This covariance matrix can be minimized by choosing K from the following equation:

$$K = PC^T R_2^{-1}$$
$$AP + PA^T + R_1 - PC^T R_2^{-1} CP = 0.$$

When choosing K from the above equation, K is called a Kalman Filter [7].

3.1.2 Fuzzy Logic

Fuzzy logic was first presented in 1965 by Lofti Zadeh as an attempt to bring more fuzzy linguistics into math. This is done by taking input variables and fuzzyify them with the help of membership functions. A membership function is a function that maps input values to membership values from zero to one. There is no limit to how many membership functions can exist, and the inputs can belong to many different memberships at once. This is the whole idea of fuzzy logic, that is instead of using boolean as in normal logic something can belong to more than one class.

From this fuzzification there will be fuzzy variables which can then be changed with a set of fuzzy rules. These rules are very linguistic and are supposed to make it easy for humans to understand what is happening.

The output from the fuzzy rules is then defuzzified with the help of another set of membership functions. How to defuzzify can be done in many ways. One method is to take how much the output belongs to each membership and find the equilibrium x-value of the resulting shape [17]. A flow chart of this whole process can be seen in Figure 3.1.

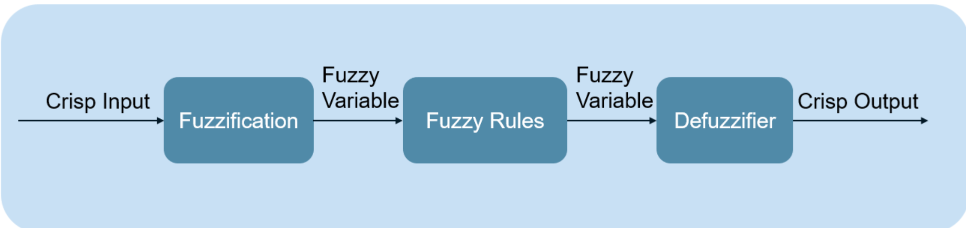


Figure 3.1: Flow chart of how a fuzzy inference system works.

3.2 Trajectory Prediction

Schubert et al. (2008) evaluated methods for motion models for a car. The authors predicted the upcoming position in a turn by assuming, for instance, that the vehicle has constant acceleration and constant yaw turn rate. The method combined GPS and odometry data with an

Unscented Kalman Filter for estimating upcoming position. The estimation was compared to a highly accurate GPS receiver and used as ground truth for the Kalman Filter to correct its predictions. The authors found good results for predicting trajectory [20]. The currently deployed warning fields for CDI at Toyota is based on a similar motion model as Schubert, but with more uncertainty in trajectory prediction. Thus, leading to a warning field that needs to take many directional cases into account and becoming very wide.

In 2018 Quehl et al. presented and evaluated other vehicle trajectory prediction methods. Quehl et al. compared one of Schubert's motion model methods of constant yaw turn rate and linear acceleration (CYRA) with their own approach based on statistical information about behaviors of other traffic participants in each area. The statistical method looked for intersections in the road network and stored vehicle velocities for many points in the road network. During trajectory prediction for an intersection, the vehicle analyzed similar speeds of other previous vehicles at this specific point and looked at which path had been taken in intersections. In tests, the statistical predictions method performed slightly better than motion model method for short distance predictions and much better at long distance predictions [15]. The tests also show that the CYRA motion model performs very poorly at long distance predictions. This weakness resembles very well with the currently deployed warning fields on CDI at TMHMS as the deployed warning fields were developed from a motion model. The motion model may be a good solution for shape and size of protective field, which only needs to look forward for a short period of time in the future. But the motion model has too much uncertainty for the future, as turn rate and acceleration is, most of the time, not constant for any AGV in a long-distance trajectory prediction. A statistical approach seems to provide more realistic results for trajectory prediction for cars and could possibly be implemented for autonomous trucks.

A very modern approach to predicting future position or "motion forecasting" has been presented by Gilles et al. in 2021 with the help of neural networks (NN). The authors trained classic convolutional networks to predict a future location for a given time horizon. The NN model took vehicle status including vehicle pose and short past vehicle trajectory for ego vehicle and neighboring vehicles, and a rasterized image structure of the road network ahead into account. This method was more complex and computationally heavy compared to previously mentioned methods as there were many input parameters. The trajectory prediction for the vehicle was then visualized with a heatmap coloring overlay, hence the name algorithm name HOME: Heatmap Output for future Motion Estimation. HOME performed very well and was applied to the Argoverse Motion Forecasting Benchmark [1] and, when published, ranked first place on the online leaderboard [6]. As of May 2022, almost one year later, HOME is ranked 16th among algorithms on Argoverse Motion Forecasting Highscore. A trained NN model performs very well for trajectory prediction and could most likely perform very well trajectory prediction for autonomous trucks. Large labeled datasets are required to train any NN model. For Gilles et al, data was not a problem as Argoverse provided the large dataset. For this thesis it was deemed that collecting a large dataset in similar size would be very cumbersome for the given time frame.

Regarding methods for trajectory prediction, this thesis method chooses to take a somewhat naive data driven method like Quehl but not as data required as Gilles. More details on the chosen method for trajectory prediction can be seen in Section 5.3.

3.3 Warning Field Generation

Since this thesis is about warning systems and how to improve the current warning fields at TMHMS it is interesting to see what methods other researchers have used and what their results are. One of these methods is the one described by Täubig et. al. where they calculate a safety zone in which they estimate the AGV will be in if it tries to stop [23]. This safety zone's shape and size is calculated with a function taking velocity v and angular velocity ω as input.

This safety zone's size is padded to account for any error in the measurement. They can then calculate an interval for the breaking distance s and the angle of the AGV regarding an origin α, see Figure 3.2. This safety zone can finally be converted to a laser scan representation [23]. The method described by Täubig et.al. gave decent results and it was the first of its kind where it looks at where the AGV will be in the future. This model is similar to the one used today at TMHMS, with a few differences such as the one at TMHMS are pre calculated and have three fields instead of one. Since they are so similar they have the same problems, such as losing precision for longer distance future location of the AGV, and thus will not be used in this thesis.

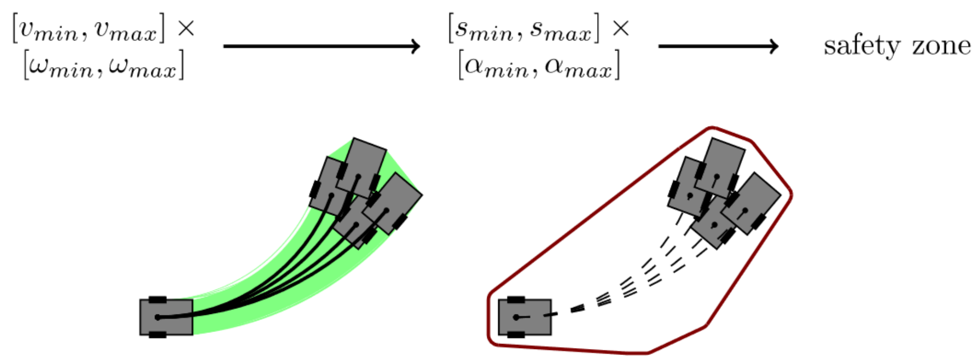


Figure 3.2: Image representation of Täubigs et. al. model. Image taken from [23].

Another method is the one by Schlegel [19]. In this method there are two parts, first there is the offline part which are pre-computations made before and then there is the online part. In the offline part computations are made for the allowed speed of the AGV if an obstacle would be in its path. These speeds are stored in a lookup table and are then indexed with the velocity v and angular velocity ω of the AGV. Schlegel then creates a circle c with radius $r(c) = v/\omega$ such that it intercepts with the AGVs center, this circles center will be called Instantaneous Center of Curvature (ICC). For the objects, o, in this circle a new circle k is created with the same radius as to the object from the ICC. If the circle k intercepts with the AGV at any point p the AGV will have to change velocity to avoid the obstacle depending on the angle α and distance d between o and p, see Figure 3.3 [19]. Schlegels method is an older method that brings out many interesting ideas. However, this method on the other hand has a few flaws. One of them is that he assumes discretized velocity and angular velocity which could slow down the AGV. It would slow down the AGV since the optimal speed in regards of efficiency could be in-between the intervals that are set. Another, and bigger, flaw is that it picks up too many objects and needs to do make many unnecessary stops as shown in the thesis by Vaidya and Bheemesh [25].

Instead, Vaidya and Bheemesh proposed in 2017 another solution where they consider the dynamics, kinematics and shape of the AGV, they also look at the future path of the machine. Their method works by widening the future path to the width of the machine. Afterwards they simply say that anything in that path is a warning, and the machine should slow down. Added to their warning field, except the path, is any part of the world that would be between the machine and the path. They do this because if they can-not see the whole path, they can-not guarantee if something is in the way on that path segment [25]. They show that their method works better than both the method from Täubig and Schlegel, which is a clear improvement. However, this is not applicable completely in this thesis since the AGV at TMHMS might not know the future path it will take. Another reason why this can-not be used is that the machines used in this thesis have a protective field which Vaidya and Bheemesh does not consider.

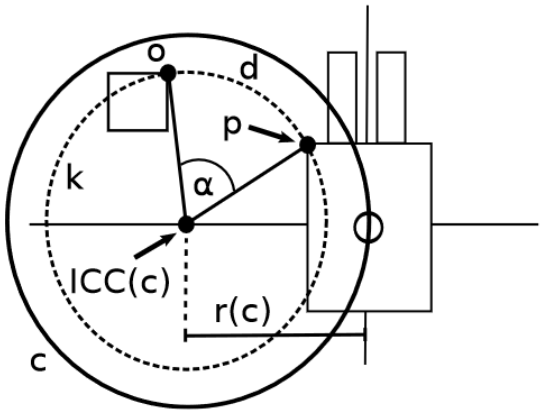


Figure 3.3: Image representation of Schlegels model. Image taken from [2].

3.4 Decision Making

A big part of this thesis is what to do when an object that is in the way is detected. The obvious answer is to slow down, but how much to slow down is another question. This is not uncharted territory and there have been a lot of studies in this field. One popular method for decision making is to use a fuzzy inference system (FIS).

For example, in 2021 Waga et. al. used a normal FIS to do obstacle avoidance with a NAO robot in unknown environments. As inputs to their system, they used two ultrasonic sensors, one to the left of the robot and one to the right. They also used the current velocity of the robot. Through fuzzification, 75 fuzzy rules and defuzzification they got an output that told the robot to either turn left, turn right, or go straight. All these options also had the possibility to keep the current velocity or increase the velocity. They could show that their system based on fuzzy logic did outperform the old method used on their robot [26]. However, their results were tested on a very simple world with only one obstacle. For this thesis that is not enough.

Another group of researchers in 2021 used fuzzy logic as an anti-collision system in cars. Here they used an ultrasonic sensor to measure the distance to objects and a humidity sensor to measure the humidity of the road to get an understanding of how good the road conditions are. As inputs to their FIS they use these two parameters as well as their own velocity. In this paper they only use 18 fuzzy rules compared to the 75 rules in [26]. What they get after these rules is an output of if they should brake or not. To test their method, they ran simulations with different values of their input values, and they could see that with their model they got the expected behavior where they brake harder if an object is closer or if they drive faster [10]. This looks good but is not perfectly suited for this thesis. The reasoning for this is that it is reliant of the road conditions which is important for vehicles driving outside where it could rain or snow, but in our case, the CDI is always inside where the road conditions may not vary as much. Another flaw is that they do not know if a turn will be happening or not. There could be cases where an object is close by, but the machine would turn anyway.

In 2015 Pothal and Parhi released a paper where they used an adaptive neuro-fuzzy inference system to navigate multiple robots in cluttered terrain. A neuro-fuzzy inference system is a combination of fuzzy logic and neural networks. In their paper their model is made from five layers where the first is a fuzzifier layer, the second and third are normal neural network hidden layers and the fourth is a form of defuzzifier, with the final layer is a simple summarization layer. The idea behind this is that you use the learning capabilities of the neural network and apply it on the FIS. On the same hand you still have the adaptability of fuzzy logic. As inputs in this method, they had the distance to obstacles in front of them, to the left and right as well as the angle to their goal position and from all of this they get a steering angle of which they should continue with. With this model they could then show good results for this type of problem where the robots always found short paths towards the goal [13]. This is a very interesting methodology that could be applicable in many topics. However, with the limited computational power on the machines used in this thesis, this method would not be possible to use in its intended frequency.

Of course, there are other solutions that are not based on fuzzy logic. An example of this is a method from 2016 based on the artificial bee colony algorithm (ABC). This ABC algorithm is an optimization algorithm where, in this case, it tries to find the optimal velocity of the vehicle. ABC works by assigning bees with random values and then evaluating these before trying to improve upon them. In their paper they used the following inputs:

- Their own velocity.
- Object's velocity.
- Distance to the object.
- Weather condition.
- Environmental data, such as how steep the road is.
- Road condition.

On the other hand, the outputs are the probability of a collision and which velocity they should have to avoid a collision. They trained the model on the NASS General Estimates System dataset, known as GES. The dataset contains many variables in records of motor vehicles crashes. With this method they showed that the ABC algorithm was able to slow down the speed of collisions and in situations where collision where non avoidable it still managed to reduce the effect of the collision [28]. This is an interesting method but could cause some troubles in this thesis. Their work looked forward for collisions, that means the algorithm looked at all objects in front of the vehicle. In the environment where the AGVs will operate, there will exist many objects that are not in the vehicles path. Thus, there is no need to take them into consideration.

Another solution is presented by Yuan et. al. in 2022 by using deep reinforcement learning with game theoretic decision making. Here they wanted to have a vehicle drive through a cross-road with another vehicle coming from each road. They set it up in such a way that there were guaranteed collisions. As input to their decision-making system, they had 2D LiDAR data and the states of each car, but first it went through a long short term memory (LSTM) cell. The different actions the machine can take are regarding velocity which are to brake, maintain or accelerate. The game's theoretical part comes with their so called Level-k reasoning. This is to combat the fact that different drivers drive differently. With this setup they can see that their model works as well as humans would handle the same crossing [29]. This is a new and exciting method, sadly it requires a lot of training to get good results, which is not suitable for this given time frame of this thesis. They also assume different agents in their world, which is not the case in this thesis.

4 Theory

In this chapter theory for some concepts and problems needed for this thesis are presented.

4.1 Coordinate Transformation

To translate coordinates from one coordinate system to another, one may use transformations from (x', y') to (x, y). These transformations look like the following:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix},$$

where θ indicates the angle of the previous coordinate system to the new coordinate system.

Converting polar coordinates to cartesian coordinates is done through:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} rcos(\theta) \\ rsin(\theta) \end{bmatrix},$$

where r is the distance and θ the angle of the vector.

4.2 Create a Circle from Three Points

There are many ways to create a circle based on three points in the plane. One way is to use the law of sine, that is $2R = A/\sin\alpha$ where R is the radius, A is the side length opposite of the angle α. Another method would be to use normal algebra to derive an equation for the radius, which can be done by starting to rewrite the equation of the circle into a more general form:

$$\begin{aligned} R^2 &= (x-a)^2 + (y-b)^2 \\ &\Leftrightarrow \\ 0 &= x^2 + y^2 + 2sx + 2ty + c, \end{aligned}$$

where $a = -s$, $b = -t$ and $r^2 = a^2 + b^2 - c$. Assume now that there are three points (x_1, y_1), (x_2, y_2) and (x_3, y_3). These points can be used as points on a circle, creating the following three equations:

$$
\begin{aligned}
0 =& x_1^2 + y_1^2 + 2sx_1 + 2ty_1 + c \\
0 =& x_2^2 + y_2^2 + 2sx_2 + 2ty_2 + c \\
0 =& x_3^2 + y_3^2 + 2sx_3 + 2ty_3 + c.
\end{aligned}
$$

From these equations it is possible to solve for s, t and c to get the following:

$$
\begin{aligned}
s &= \frac{(x_1^2 - x_3^2)(y_1 - x_2) + (y_1^2 - y_3^2)(y_1 - y_2) + (x_2^2 - x_1^2)(y_1 - y_3) + (y_2^2 - y_1^2)(y_1 - y_3)}{2 * ((x_3 - x_1)(y_1 - y_2) - (x_2 - x_1)(y_1 - y_3))} \\
t &= \frac{(x_1^2 - x_3^2)(x_1 - x_2) + (y_1^2 - y_3^2)(x_1 - x_2) + (x_2^2 - x_1^2)(x_1 - x_3) + (y_2^2 - y_1^2)(x_1 - x_3)}{2 * ((y_3 - y_1)(x_1 - x_2) - (y_2 - y_1)(x_1 - x_3))} \\
c &= -x_1^2 - y_1^2 - 2sx_1 - 2ty_1.
\end{aligned}
$$

Finally, the radius can be computed as:

$$R = \sqrt{s^2 + t^2 - c}.$$

Both methods, the law of sine method and the algebraic method, have the same time complexity and approximately the same space complexity.

4.3 Deciding Wheel Velocity Based on Turn Radius

To predict what type of protective field (as described in Section 2.5) the machine will have in a future location it is desired to know the wheel velocities in that location. These velocities can be derived by basic kinematics because the AGV in this thesis is a differential driven machine. It is known that the angular velocity ω is a quotient of the velocity V and the radius R, that is:

$$\omega = \frac{V}{R}.$$

If the machine is in a turn the left wheel will have a turn radius of R_l while the right wheels will have a turn radius of R_r. These are correlated in such a way that the difference between R_l and R_r will be the distance between the wheels, W [4]. All of this together comes together as follows:

$$
\begin{aligned}
\omega &= \frac{V_l}{R_l} = \frac{V_r}{R_r} \\
R_r &= R_l + W \\
\omega &= \frac{V_r - V_l}{W}.
\end{aligned}
$$

Since the velocity V the machine will be going is the average of both wheel velocities it is possible to rewrite this as:

$$R = \frac{V_r + V_l}{2} \cdot \frac{W}{V_r - V_l}.$$

Finally, the wheel velocities can be calculated according to the following equation:

$$V_l = \omega(R - W/2) \quad V_r = \omega(R + W/2). \tag{4.1}$$

4.4 Orientation of Three Points

The orientation of three different points in the plane is a way to describe how the points are oriented and can have three different values. Either it is clockwise (right turn), counterclockwise (left turn), or they are collinear. To decide the orientation of these points one can look at the slope between point a and point b, and compare that with the slope from point b to point c. The slope, σ, from a to b can be calculated as:

$$\sigma = \frac{b_y - a_y}{b_x - a_x},$$

and for the slope, ϕ, between b and c it would be:

$$\phi = \frac{c_y - b_y}{c_x - b_x}.$$

If $\sigma > \phi$ it would be a right turn otherwise it would be a left turn. As the orientation is dependent on the sizes of the slope it is possible to rewrite this equation into equation 4.2. If o is larger the 0 it would be a right turn otherwise it would be a left turn. If o would happen to be 0 the points would all lie on the same line (they would be collinear) [5]. Examples are shown in Figure 4.1.

$$o = (b_y - a_y)(c_x - b_x) - (c_y - b_y)(b_x - a_x) \tag{4.2}$$

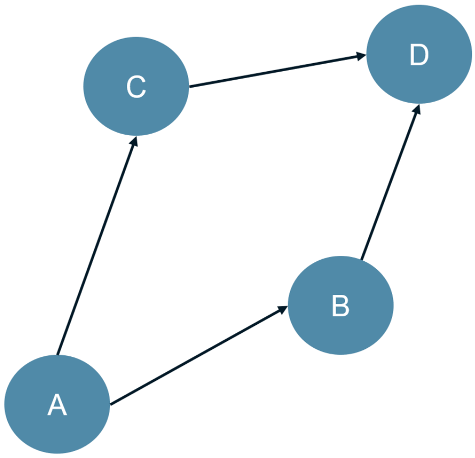


Figure 4.1: Points A, B and D have a counterclockwise orientation, while points A, C and D have a clockwise orientation.

4.5 Line Intersection

To determine if two line segments, one with endpoints (a, b) and one with endpoints (c, d), are intersecting one can use the orientation of the points. There are two cases, one generic and one special case. For the generic case it states that if the orientation of a, b and c, and a, b and d are different, and if the orientation of c, d and a, and c, d and b are different the lines intersect. The special case states that if any of the orientations are collinear and the point that does not belong to that line segment lies on the line segment they intersect [5]. Examples of this can be seen in Figure 4.2.

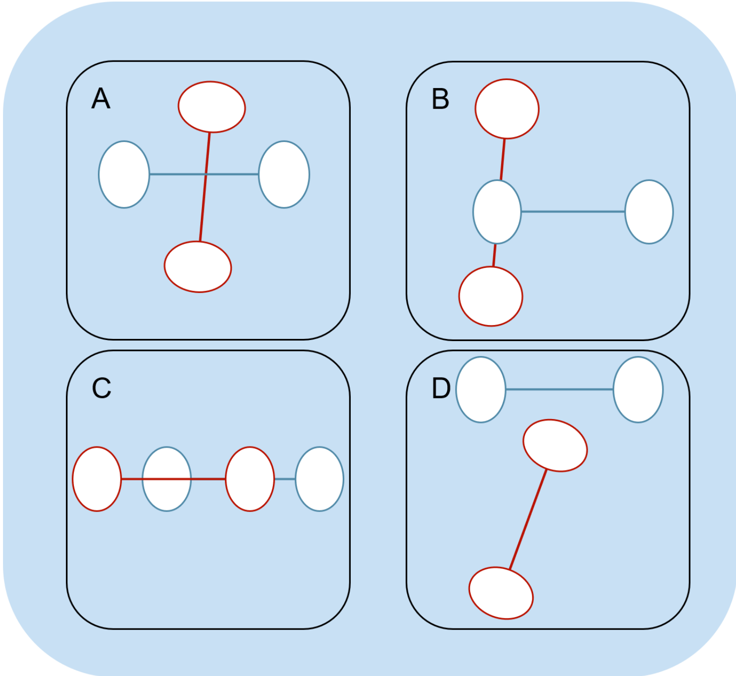


Figure 4.2: In cases A, B, and C the lines intersect in three different ways, where case A, and case B are the generic case and case C is the special case. In case D the lines do not intersect.

4.6 Detecting Objects Inside a Polygon

How to detect if an object is inside a polygon is a big problem both in this thesis and in general regarding autonomous driving [30], but it is also a problem in other areas such as geographical systems [12].

4.6.1 Ray-Casting Algorithm

One way to solve this problem would be with the ray-casting algorithm first presented in 1962 [21]. This algorithm works by drawing a horizontal line from the point to infinity. One can then count the number of intersections between this new line and the polygon. If the number of intersections is odd, it means that the point is inside the polygon. If it on the other hand is even it is outside the polygon, an example of how this would work can be seen in Figure 4.3. One problem with this algorithm is the case when the line intersects the polygon in an edge since it would be counted twice. This could be solved by not counting the vertices more than once. The ray-casting algorithm has a time complexity of O(n) where n is the number of vertices in the polygon.

4.6.2 Winding Number Algorithm

Another solution to the problem of a point inside a polygon is to use the points winding number, that is how many revolutions one can make around the point while traveling along the polygon's edges. If the winding number ω would be zero, then the point is outside the polygon otherwise it is inside [8].

There are many ways to calculate the winding number, one way would be to use an incremental angle algorithm as described in [27]. With this algorithm one sums up the angles between the point and each edge. The summation would then add up to $\omega \cdot 2\pi$. An example of this can be seen in Figure 4.4. This way of testing if the point is inside or not would have a

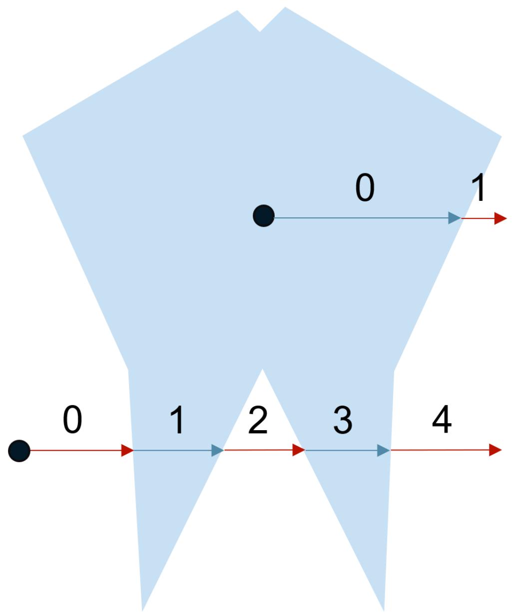


Figure 4.3: Ray-casting algorithm counts the number of intersections with the polygon.

time complexity of O(n) where n is the number of vertices. However, it would include a lot of inverse trigonometric functions.

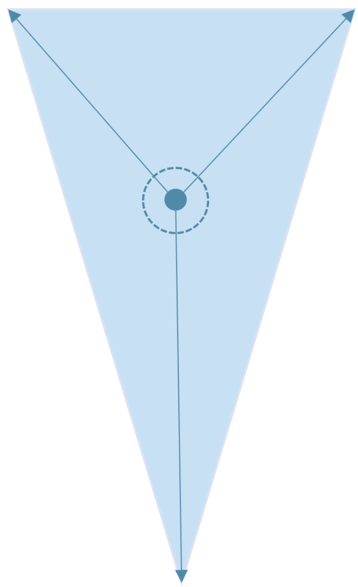


Figure 4.4: Winding number algorithm counts the amount revolution around the point. One way to do this is with the angles to the vertices.

4.6.3 Polar Coordinate Solution

A third simpler solution to this problem would be to describe the polygon as polar coordinates regarding one point on the polygon. Then instead of a number of vertices there would be a number of angles and lengths at each angle. It would then be trivial to look at an angle and check if the distance is shorter or not at that angle. If the distance to the point is shorter than that of the polygon the point would be inside otherwise it would be outside. However, this method has a few limitations. One of the biggest flaws with this is that it will not be as exact as the previous methods since there would have to be a finite amount of angles and it could not be continuous. Another flaw is that it is not general, meaning for this method to work each angle may only have one length. A last downside, that is not such a bad thing in this thesis, is that the polar coordinates need to have an origin on the polygon, and the choice of origin is important since different locations of origin may give different results. This would not be a problem for this thesis since the origin would be the scanner's location.

5 Method

This chapter revolves around the methods used to solve the research questions. That is:

1. How can a predictive warning system based on empirical data be implemented on a warehouse AGV at TMHMS?
2. How would such a warning system, compared to the current warning system at TMHMS, perform in the following aspects:
   a) *Driving performance:* Time to finish a short industry-like test track?
   b) *Driving dynamics:* Perceived safety, driving smoothness and overall quality?

To evaluate driving dynamics and performance for a warning system based on empirical data, the system needs to be implemented first. Therefore, the method section will explain the development of such a system. The developed warning system was an embedded program running on a CDI together with other necessary software handling communication, navigation and safety on the CDI. In a simple format, the program works in the following manner. (1) While driving, the program continuously stores past driven coordinates and their corresponding amount of occurred vehicle headings. Given time, the CDI will have a decent statistical understanding of what paths lead to which places. (2) The program estimates a future position by iterating over a set number of coordinates in front of the CDI to a fixed distance. (3) For every coordinate in the iteration, the program simulates how the CDIs protective field will look in this coordinate. As the CDI's protective field's size is dependent on speed, the program will simulate different speeds for all iterated coordinates. (4) If objects are inside of a future simulated protective field, the program will choose a low enough speed to enable passing of the object outside of the protective field, if possible. A general system overview can be found in Figure 5.1. Besides the general system overview one can zoom into the Predictive Warning System block and see a flowchart in Figure 5.2.
The program has been developed in the programming language C++ and ran separately on the internal computer of the CDI. The following sections will provide deeper explanations of the subsystems in the program.

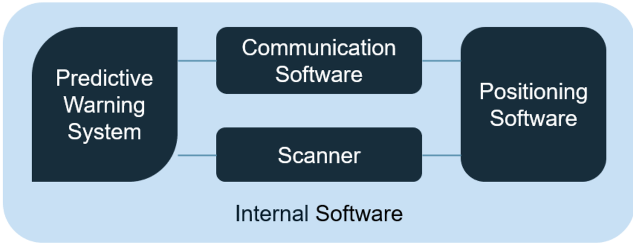


Figure 5.1: A system overview of the developed warning system. Notice the new warning system.

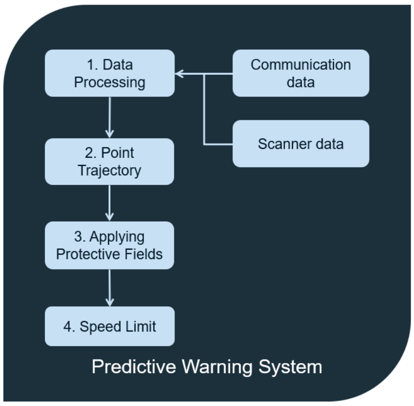


Figure 5.2: A flowchart of the system.

5.1 Collecting and Processing Scanner Data

Onboard the CDI we find the LiDAR scanner which can be, as explained in Section 2.4, communicated with the TCP/IP protocol. When the code receives the data from the scanner it was in byte format which were translated to something more understandable for the user (such as in base-ten). The data will then be on the format $[distance, beam\_nr]$. Here the *distance* means the distance to the hit and *beam_nr* is which beam that distance belongs to. This is not in any normal coordinate system (such as cartesian or polar). Converting the data to polar coordinates equation 5.1 is used. In this equation *start_angle* is the first beam's angle, $-47.5°$, and *resolution* is the angle between each beam. After this equation the scanner data will be in the format $[distance, \theta]$.

$$\theta = start\_angle + beam\_nr \cdot resolution \tag{5.1}$$

Afterwards, the scan data would have to be converted to cartesian coordinates in the global world the machine is operating in. First the data can be converted to cartesian coordinates with origin at the machine center, see Figure 2.1b. This can be done with equation 5.2, where x_l and y_l are the local coordinates and $scanner_x$ and $scanner_y$ is the scanner's location on the machine.

$$\begin{bmatrix} x_l \\ y_l \end{bmatrix} = distance \cdot \begin{bmatrix} \cos(\theta - \frac{\pi}{2}) \\ \sin(\theta - \frac{\pi}{2}) \end{bmatrix} + \begin{bmatrix} scanner_x \\ scanner_y \end{bmatrix}. \tag{5.2}$$

In this thesis the scanners coordinates (in meters) are:

$$\begin{bmatrix} scanner_x \\ scanner_y \end{bmatrix} = \begin{bmatrix} 0.52 \\ 0 \end{bmatrix}$$

From this it is possible to get these coordinates in the global world with the help of coordinate transformation, as explained in Section 4.1, and then shifting this to where the machine is in the world. The exact equation can be seen in equation 5.1.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} machine_x \\ machine_y \end{bmatrix} + \begin{bmatrix} x_l \\ y_l \end{bmatrix} \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix}$$

5.2 Storing Vehicle Poses

The communication module provides vehicle poses which are vehicle coordinates and vehicle headings. These vehicle poses was saved in a map, that is, a grid map for all driven coordinates. The resolution of the grid was set to decimeters to have a balance between cost and detail. A data container in the form of a hashmap was created with coordinates as keys. Each key will then map to another hashmap with different headings as keys at that coordinate. These headings will be rounded to an integer between 0 and 360 and then map to the number of occurrences of that specific heading and what coordinate the middle of the machine had at that heading. This data object is visualized in Figure 5.3. Why some of this data is needed may seem unclear, but the data will be used for predicting the future path in Section 5.3. This map object is updated each time the machine reaches a new grid position from the previous location (the grid distance is a variable that can be changed to change the resolution of the map). When it updates it does not only update its own location, which is the middle point of the CDI. Instead, it updates points in a line along the y-axis over the middle point to get the whole width of the machine. See Figure 2.1b for direction of y-axis. An update is to increment the heading values in all coordinate points on the middle line.

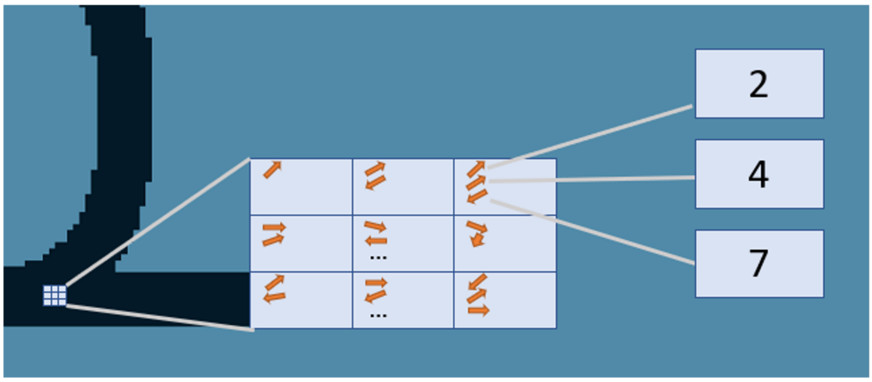


Figure 5.3: Visualization of the data object storing vehicle poses for all driven coordinates

Since the outer hashmap has a tuple (the location in x and y coordinates) as a key, a custom hash function is needed. This is done by using the bitwise xor function between the x and y values. This will guarantee one of the requirements of hash functions, which is that it should be hard to find two inputs, m_1 and m_2, to which the hash function returns the same hash.

5.3 Trajectory Prediction

By looking at previous driving examples from the vehicles current position, the vehicle can estimate where it will be driving in the future. This method will first start in the vehicle's current position with its current heading, h_n. It will then step forward a fixed number of meters, d, in the headings direction. This step is calculated with the following equation:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \end{bmatrix} + d \cdot \begin{bmatrix} \cos(h_n) \\ \sin(h_n) \end{bmatrix}, \tag{5.3}$$

where n is the iteration count which is a predetermined integer. These new coordinates are then controlled to see if they exist in the map object. If they exist in the map object, a new heading will be chosen. The new heading will be chosen by finding the most common heading at the new locations position. However, this heading also has to be similar to the previous heading. The reasoning for this is because it is assumed that the machine will not go in the opposite direction that it came from. With this new heading it will once again iterate forward to a new position and the process is then repeated n amount of times.

While iterating forward the predicted position does not exist in the map object the algorithm would take one step back on its predicted path and continue with the second highest heading and so on. If for some reason it would not find a solution, it will assume it is driving in a new location and thus drives very slowly. This is written in pseudo-code in Algorithm 1 and a sketch of the predicted outcome can be seen in Figure 5.4.

Algorithm 1 Algorithm to predict a future path

Input: H_c the number of headings, M map with saved data, T_p list of trajectory points to fill, T_c how many trajectory points should exist
Output: Boolean if a path was found

```
function FUTUREPATH(H_c, M, T_p, T_c)
    if T_c == length(T_p) then
        return True
    end if
    (x, y) ← Predicted location                ▷ Calculated with equation 5.3
    p_l ← (x, y)
    if p_l ∉ M then
        return False
    end if
    for H ∈ M[p_l] do                          ▷ Loop through each heading
        C_h ← H − P_h        ▷ Difference between heading and previous points heading
    end for
    sort(C_h)                                  ▷ Sorts by ascending order
    for i ∈ min(H_c, length(C_h)) do
        A ← C_h[i]                             ▷ Get the angle at location i
        C ← (M[p_l][A], A)
    end for
    sort(C)                                    ▷ Sorts by descending order
    for c ∈ C do
        L ← (x, y, c[1])
        T_p ← L                  ▷ Add the new location to list of locations
        if FuturePath(H_c, M, T_p, T_c) then
            return True
        end if
    end for
    return False
end function
```

5.4 Predicting Future Protective Fields

Since the choice of protective field is made with the help of the wheel speeds on the CDI it is required to also predict the wheel speeds at each of the predicted locations. This can be done by creating a circle with three of the points, as described in Section 4.2. The two methods were presented to create a circle from three points, one with the law of sine and one algebraically. In this thesis, algebraically will be used since none of the information for the other method is known earlier and would have to be computed. From that circle the radius can be used to

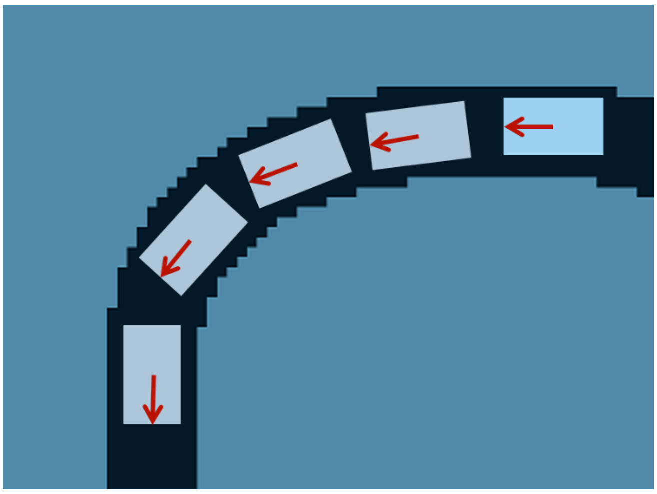


Figure 5.4: Visualization of the expected prediction of future positions based on the map data object described in Section 5.2.

calculate the wheel speeds as described in Section 4.3. From this a radius can be determined, but for equation 4.1 to work properly a negative radius must be determined. Another thing that has to be decided is a max value for the radius (since if the three points are collinear the radius would be infinite), this max value will be $100,000$. The negative radius will appear if the machine is making a right turn (since V_l would be larger than V_r). To find if it is a right or left turn the orientation (as described in Section 4.4), o, of the three points can be calculated with equation 4.2. With this radius, the wheel velocities can be calculated with equation 4.1.

When the wheel velocities have been calculated a protective field, as described in Section 2.5, can be predicted with the help of a lookup table from TMHMS. Here the field is divided into different speed intervals. Each wheel's velocity will get its interval and depending on the difference between the intervals the direction of the field will be chosen. When the direction is chosen the average velocity of the two wheels gets its interval and this will decide the field.

5.5 Deciding Suitable Speed to Eliminate Protective Stop

When applying the protective fields to the predicted points the first thing that happens is that the current velocity is set as the theoretical maximum speed (TMS). The TMS is how fast the CDI is allowed to go without getting a *protective stop*. This value can both increase and decrease. Afterwards the first three points of the estimated path are used to get a field for the current TMS. This field is then applied to the two first points. When the field is applied to a point it looks if any object from the scanner data is inside its field. This is done with the ray-casting algorithm, as explained in Section 4.6. The reason for this algorithm over the others is because it is well documented and that it is easy to implement. If there would be any scanner point inside the estimated locations field, it would decrease the TMS enough for the *protective field* to go down one interval. When the speed has been lowered it tries again to apply the field on that point. This process is repeated until it finds a speed that would not trigger a *protective stop*. On the other hand, if the field would not contain any point it would try to increase the TMS enough for the *protective field* to go up an interval. It would then recompute the field and apply it again. If possible, it would increase until it cannot increase more.

When the first point on the estimated path has approved a speed, it is the next points turn to apply the same field. Once again it will lower the TMS until it finds a good speed that does not trigger the *protective field*. When the two first points from the estimated path have approved a speed, a new field will be generated with points number two, three, and four. This new field will only be applied to the point number three. This field will then lower the

TMS like the others if needed. This process is then repeated until all of the estimated location points have a field. When there is only one estimated location point left it will be included with the second and third to last points. Note that only the first point can increase the TMS, since each field is bigger than the field corresponding to one interval below it. This creates a scenario where if one estimated location point approved would approve one TMS, all lower values for the TMS would also be approved.

If it happens that a estimated location point cannot find a field that can get past an object it will set the TMS to 25·field_number mm/s, where field_number is the point where the field is applied starting from 0 up to the set number of points. This is because it is desired that the CDI will go really slow towards the object and stand still close to it instead of standing still far away

When every estimated location point has gotten a field, the TMS will be the speed limit sent to the communication software. This algorithm can be seen in pseudo-code in Algorithm 2.

5.6 Testing and Evaluation

In Figure 5.5 an illustration of the test track can be found. The test track was developed to be as simple and short as possible, while still provoking the warning systems in challenging segments. The segments were created from discussions with TMHMS and the segments were based on realistic and challenging driving examples for a CDI. The CDI will then drive this track with the different warning systems for two laps for each warning system. The time it takes for the CDI to finish a lap is then noted.

The CDI follows the track in a counter-clockwise-orientation and collects all three checkpoints from one to three. The checkpoint's locations can be seen in Figure 5.5. The track is divided into three segments for future analysis and easier naming referencing. The three segments are directly correlated to the challenging passages mentioned before.

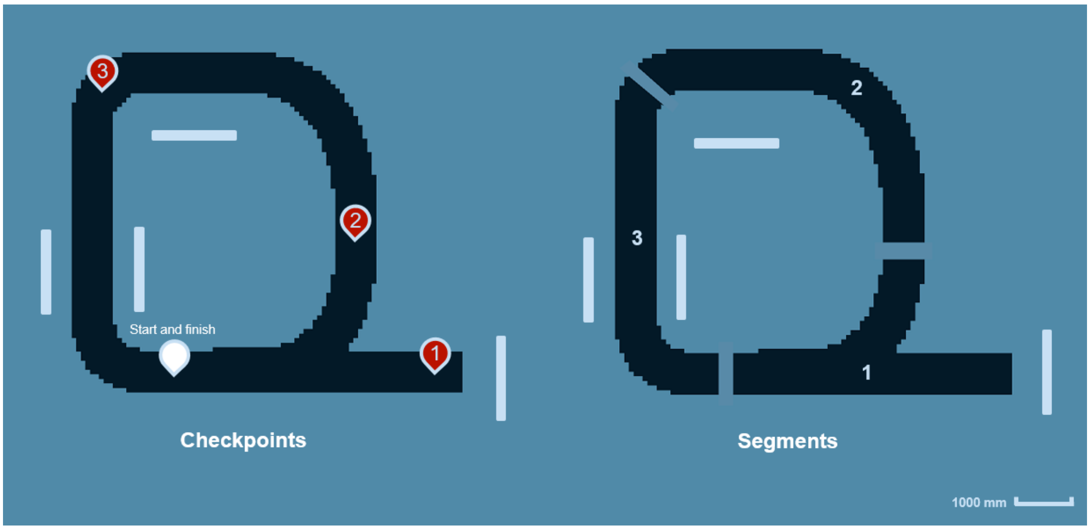


Figure 5.5: Test track for the CDI to drive in while comparing warning systems.

In the following subsections segment's shape and purposes will be presented.

5.6.1 Segment 1

In the first segment, the CDI is informed to drive with high speed straight in the direction of a wall to collecting Checkpoint 1. Here it is of interest to see the warning system's ability to break accordingly in advance. TMHMS has mentioned two problems that can occur in

Algorithm 2 Algorithm to decide velocity

Input: T_p list of trajectory points, P_f list of different protective fields layout, S the scanner data

Output: S_l a speed limit

```
function UPDATEVELOCITY(T_p)
    CS_l ← current speed
    i ← 0
    for p_{i-1}, p_i, p_{i+1} ∈ T_p do
        P ← p_{i-1}, p_i, p_{i+1}
        W_v ← WheelVelocities(CS_l, P)                      ▷ Calculated with equation 4.1
        CS_l ← UpdateVelocityForAPoint(CS_l, P, P_f, S, W_v)
        if CS_l = 0 then
            S_l ← 25 · i
            return S_l
        end if
        i = i + 1
    end for
end function

function UPDATEVELOCITYFORAPOINT(CS_l, P, P_f, S, W_v)
    while True do
        Pf ← FindProtectiveField(CS_l, P_f, W_v)            ▷ As explained by Section 5.4
        P_i ← false
        for s ∈ S do
            I ← PointInsideField(Pf, s)                      ▷ As explained by Section 4.6
            if I then
                CS_l decreases one interval
                if CS_l ⩽ 0 then
                    return 0
                else if increasing then
                    return CS_l
                end if
                P_i ← true
                W_v ← WheelVelocities(CS_l, P)
                break
            end if
        end for
        if !P_i and P[0] is the first point then
            CS_l increases one interval
            W_v ← WheelVelocities(CS_l, P)
            increasing ← true
        else if !P_i then
            return CS_l
        end if
    end while
end function
```

similar zones to this segment. Firstly, without a warning system the CDI may have been too high a speed to reach the checkpoint before ordering a protective stop. Secondly, TMHMS says that some warning systems will prevent the CDI from reaching the checkpoint, because the checkpoint is too close to the wall. With this information, segment 1 seemed like a suitable challenge for all warning systems.

5.6.2 Segment 2

The second segment, challenges the warning system's ability to estimate future position while in a turn. TMHMS had mentioned that the Current Warning System (CWS) makes the CDI slow down when the turn is ending and an object, like a wall, is situated after the turn but not on the path. When the CDI is at the end of the turn, see segment 2 on the right side of Figure 5.5, its warning fields will be oriented to the left. The position of the white obstacle in segment 2 was chosen by trial and error. What was sought after was a position which made the CWS slow down, even though there is no real threat for the CDI. For a warning system capable of authentic trajectory prediction, the system would not slow down from this specific obstacle inside segment 2, as it is clearly outside of the path. Because of this hypothesis, this segment is of great interest for the warning systems.

5.6.3 Segment 3

For the last segment, the warning system's ability to maintain a suitable speed was studied. As explained in Section 2.5 and further shown in Figure 2.5, there is a problem with CWS ability to drive in a narrow aisle. This problem will now be explained in more detail. The distance between the barriers in segment 3 was chosen to be between 179-191 cm. In this distance interval we can observe the CDI having the choppy driving behavior for CWS. The distance was measured from observing the defined width of today's warning field, which can be seen in Figure 5.6.

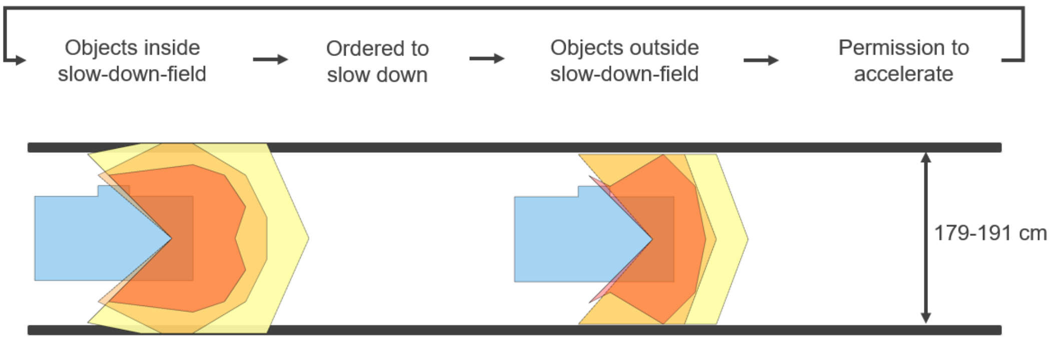


Figure 5.6: Narrow aisle problem that was expected to occur in segment C.

To the left side of the figure, the wall is inside the orange field (the *slow-down field* as explained in Section 2.5). As the name suggests, when objects enter this field, the CDI will begin to decrease in speed. By decreasing the CDI in speed, a new set of narrower warning fields will take its place. Therefore, leading to the walls not being inside of any warning field. When nothing is inside of any field, the CDI has permission to accelerate. Accelerating the CDI will in turn deploy new wider warning fields. The wider warning fields may now detect the wall once again, and an infinite decision loop has occurred. This decision loop was expected to be visible for the CDI when driving in segment C. The unwanted decision loop cannot be found in all transitions between CWS fields but only from the *stand-still fields* and the *straight 1 fields* (which is the fields for the slowest speed interval while heading straight). When the CDI drives in speeds between 0-20 mm/s the *stand-still fields* will be enabled and when driving in speeds between 21- $\approx$ 300 mm/s the *straight 1 fields* will be enabled. For the unwanted decision loop to take place the CDI must be driving in a path segment that requests the CDI to drive in speeds in the same interval as the *straight 1 fields* is enabled. For segment C a suitable speed limit of 270 mm/s was chosen for the path segment.

5.6.4 Jury Evaluation

To get an understanding of driving dynamics, developers at TMHMS were asked to observe the CDI driving the test track with each warning system enabled at a time. Moving forward these developers will be referred to as the jury. The jury consists of six TMHMS software developers that all had deep experience with driving autonomous trucks with warning systems. Three jury members' primary work was software for CDI, two other jury members' primary work was for CWS and they had a deep understanding of warning systems and protective fields for many autonomous trucks. The final jury member's primary work was in navigation for large autonomous trucks and this jury member was familiar with warning systems and protective fields. The evaluation was done by letting the jury know beforehand which warning system was active and then letting the CDI run the track two times. Afterwards, the warning system was switched to another and then finally the last one was tested. The tests were done in this way so the jury could see the different warning systems multiple times and thus give good comments about the different methods. After the jury had observed the different warning system drive the test track, they were asked to fill in a form containing three grading scales and two free text questions. The complete form can be found in Appendix A. In the form the jury was asked to rate the dynamics from one to ten in the following three metrics: perceived safety, driving smoothness and overall quality. The one to ten scale was chosen because it gave the opportunity for the jury members to be more nuanced in their rating compared to for example a one to five scale. For the three different metrics, they were all chosen both because TMHMS says that these are important factors, but also because they all show something different about machine driving. For example, perceived safety is important since the warning system does not handle safety the machine will be safe either way. But if there is a machine that does not feel safe, humans who work with the machine would be afraid and try to avoid the machine. Smoothness is important mainly to convince humans that the machine is stable and that it does what it is supposed to do, but also to lower the wear on the hardware. After the rating section two free text boxes were presented with titles *Strengths of this warning system* and *Weaknesses of this warning system*. In these text boxes the jury could respond with comments and analysis of the three systems.

5.6.5 Evaluation Method of Results

All warning systems will have its time to finish the test track measured twice to prevent randomness. The average of the two laps will then be presented and further discussed. Together with time to finish, each system's time to complete individual segments will be measured and calculated into average in the same manner. The segmental times can be of great interest to see how each system performs on different challenges. The jury's answers provided a unique and insightful rating on the warning systems' driving dynamics. Dynamics are somewhat subjective but as the jury had deep knowledge of autonomous trucks, like the CDI, they were the most suitable evaluators for driving dynamics. From the test tracks average times, together with the jury evaluation, it can be concluded which system performs with greatest dynamics and performance on the test track.

6 Results

This chapter contains results from developing a predictive warning system and evaluating the driving performance and quality of the warning systems. Firstly, results from usage of the trajectory prediction component and the speed regulation component will be presented. Secondly, results from the test track will be presented and respective warning systems time to finish the track will be shown. Lastly, a subjective evaluation of respective warning systems from a jury of TMHMS will be given from observing each warning system control the CDI on the test track.

6.1 Trajectory Prediction with Heading

This section presents results of upcoming trajectory.

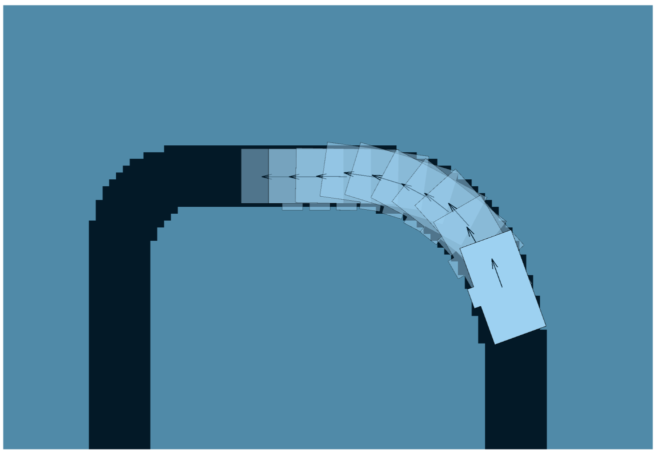


Figure 6.1: Trajectory prediction for PWS from algorithm provided in 5.3.

Figure 6.1 was generated by first letting PWS write predicted coordinates to a file during driving. Then MATLAB read from this file to visualize the predicted path. The light blue boxes represent the CDI, and its predicted future positions and headings. The black arrow represents the direction of travel for the CDI. The thick dark blue line is the true path the CDI is driving on. As seen in the figure, the algorithm predicts positions very well on the path.

6.2 Predicting Future Protective Fields

For each future location provided in Figure 6.1 PWS estimates each location's protective field, based on how sharp the turn is. The predictions based on estimated wheel speed difference from turn sharpness can be seen in Figure 6.2. The light blue future protective fields are angled in same direction as of the turn, which is correct.

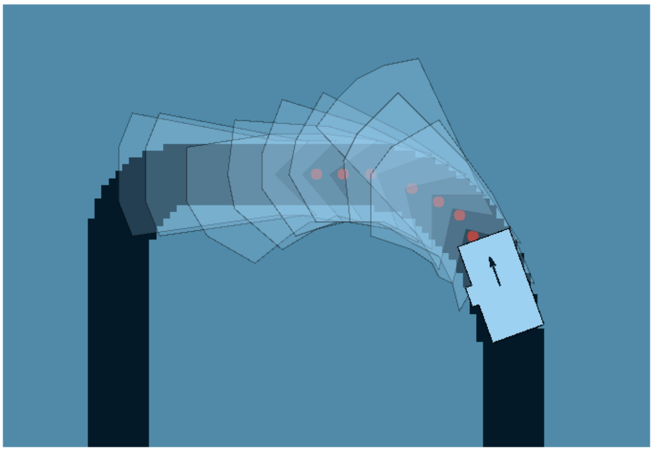
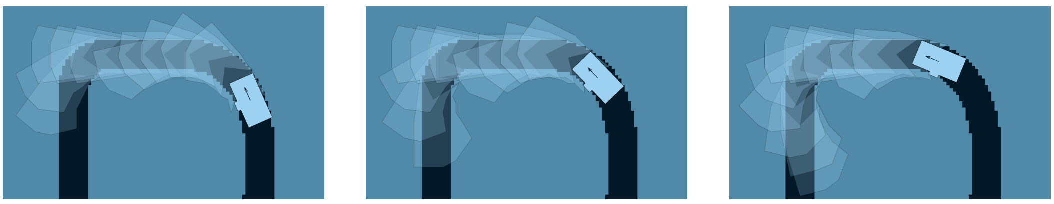


Figure 6.2: PWS estimating multiple future protective fields from methods provided in 5.4. Many of the fields are angled to the left, which is the same direction as the turn.

6.3 Deciding Suitable Speed Based on Objects in Path

In Figure 6.3 PWS was activated and the three sub figures show a time step during driving. In Figure 6.3c, all fields are at full length which tells that the CDI was granted to drive in high speed. For this scenario PWS was not limiting the speed, but instead permitted the CDI to accelerate going forward with a very high speed. Important to note is that raising a speed limit may not accelerate the CDI right away, as multiple internal modules in CDI request speed limits and the lowest requested speed limit are chosen. In this case the speed was already at the highest possible speed for this section of the path. As PWS was not limiting the speed in any way, the speed regulator in PWS was doing its intended job.



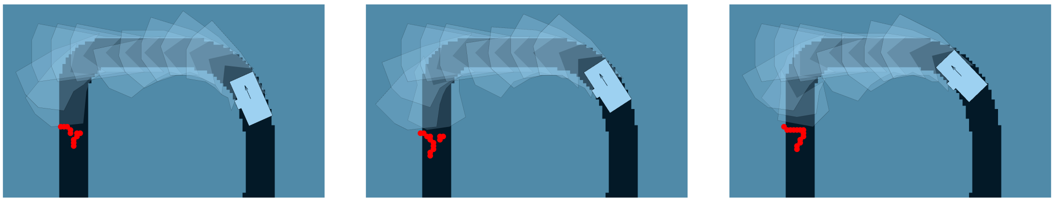
(a) Time step 1, PWS requested very high speed limit.

(b) Time step 2, PWS requested very high speed limit.

(c) Time step 3, PWS requested very high speed limit.

Figure 6.3: Three-time steps of CDI driving the test track with PWS having no object on the path.

For the next figure, Figure 6.4, the same time steps are measured as Figure 6.3 but instead, a person was walking across the intended path. The student gets noticed by the scanner and can be seen as red dots on the dark blue path. First, by comparing 6.3a to 6.4a it can be seen that two smaller future protective fields have been chosen at the end of the path closest to the detected objects. This is an attempt to avoid detected objects by reducing the speed of the CDI. Further, in time step 2 in Figure 6.4b, the CDI has moved slightly. This comes from the reduced speed PWS requested. Lastly, in time step 3 in Figure 6.4c the CDI was driving at a very low speed. The outer most future protective field was in its smallest size, that is the *Standstill field*. This means that the PWS has simulated all available lower speeds to pass the object, but no speed seemed suitable to pass the object without triggering a protective stop in the future. When this scenario occurs, PWS should request a very low speed, which is exactly what can be see in time step 3.



(a) Time step 1, PWS requested high speed limit. (b) Time step 2, PWS requested medium speed limit. (c) Time step 3, PWS requested low speed limit.

Figure 6.4: Three-time steps of CDI driving the test track with PWS having object on the path.

6.4 Driving Performance on Test Track

In Table 6.1, the average time over two laps to finish each segment is presented for the three warning systems.

Table 6.1: Average time to finish each segment on test track. A lower number indicates a faster route.

Segmental time to finish (seconds)			
Warning system	Segment 1	Segment 2	Segment 3
No Warning System	36.78	15.81	18.05
Predictive Warning System	33.68	16.19	17.73
Current Warning System	33.90	18.12	27.32

To better see which segments were challenging for each warning system Table 6.2 was generated from test track measurement. In this table the average time to reach checkpoints is presented, which also can be seen graphically in Figure 6.5. The warning system having the fastest time to finish was PWS with being on average three seconds faster than NWS. CWS finished the test track in 79.34 seconds being the slowest of the systems, around ten seconds slower than NWS.

Table 6.2: Detailed checkpoint timestamps for CDI driving with different warning systems.

Test track timestamps at checkpoints (seconds)					
Warning system	Start	CP1	CP2	CP3	Finish
No Warning System	0.00	12.00	36.78	52.59	70.64
Predictive Warning System	0.00	9.08	33.68	49.87	67.60
Current Warning System	0.00	8.86	33.90	52.02	79.34

In this table we can see a clear difference between NWS performing slowest in segment 1 and CWS performing slowest in segment 2 and segment 3. It is also of interest to see how the CDI performs while in the narrow aisle section of segment 3. In Table 6.3 it can see each warning system's time to pass one meter of narrow aisle. The amount of protective stops taking place during one meter of narrow aisle can also be seen in the same table.

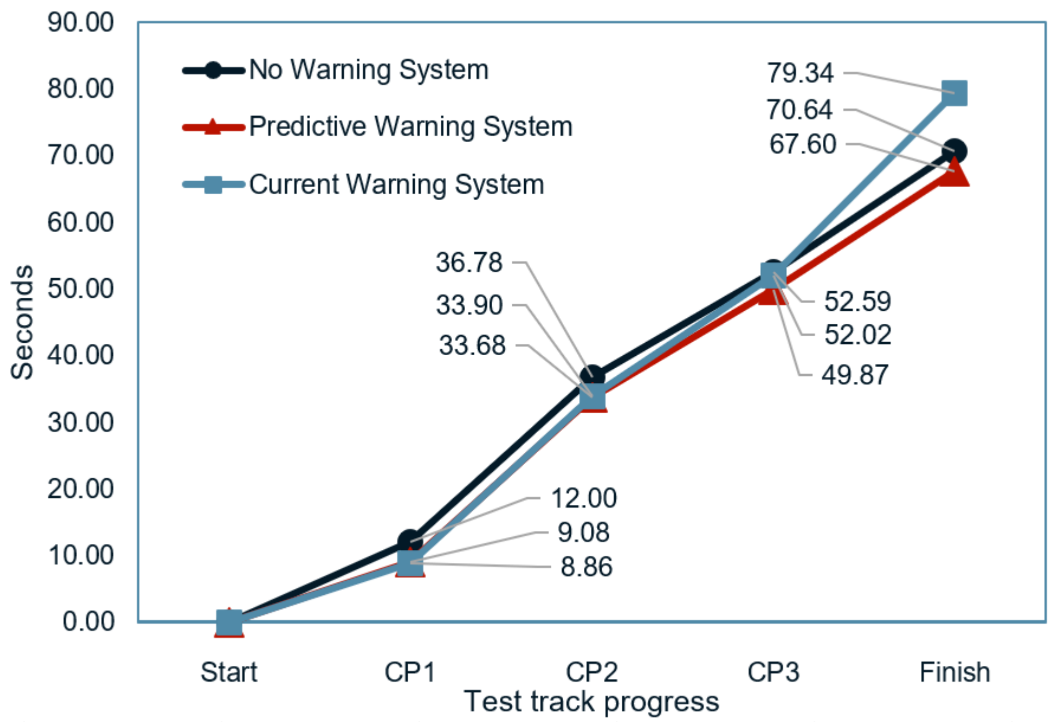


Figure 6.5: Visualization of test track race for different warning systems.

Table 6.3: Driving analysis of one meter distance in narrow aisle for different warning systems. Lower time indicates a higher average velocity.

One meter of narrow aisle of segment 3 analysis		
Warning system	Time to cover (seconds)	Amount of protective stops
No Warning System	3.5	0
Predictive Warning System	3.5	0
Current Warning System	10	10

6.5 TMHMS Jury's Rating and Comments

In Figure 6.6, TMHMS jury's rating on the three metrics connected to driving quality can be seen. In this figure, each metric is rated from 0 to 10 based on the CDIs driving having each warning system enabled, NWS, PWS or CWS.

Figure 6.7 presents same data as Figure 6.6 but provides a more compact visualization of driving quality. The plot is manually unlabeled as it should only serve as support for Figure 6.6. Here it becomes clear that PWS generated the highest combined rating compared to NWS and CWS. Swedish quotes have been freely translated to English.

6.5.1 Comments from Jury

In the following sections the free text answers from the jury have been grouped according to each warning system. Grammar corrections have been applied and some extra context has been provided with parenthesis from the quotes.

No Warning System

Two examples of strengths from NWS were, "Pretty smooth driving when not triggering prot(ective) stop" and less likely to stop without reason. Example weaknesses of NWS were

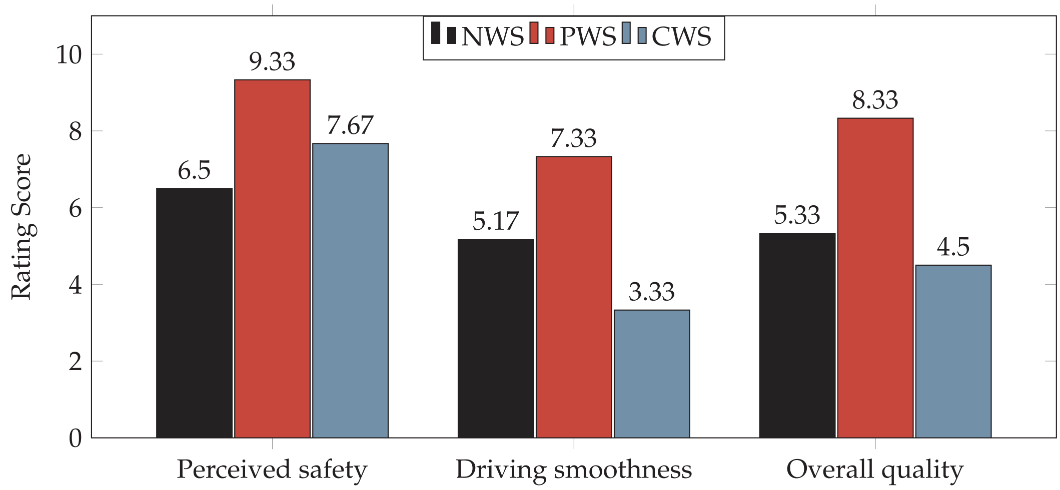


Figure 6.6: Jury's average rating on driving dynamics metrics after driving the test track with each warning system enabled.

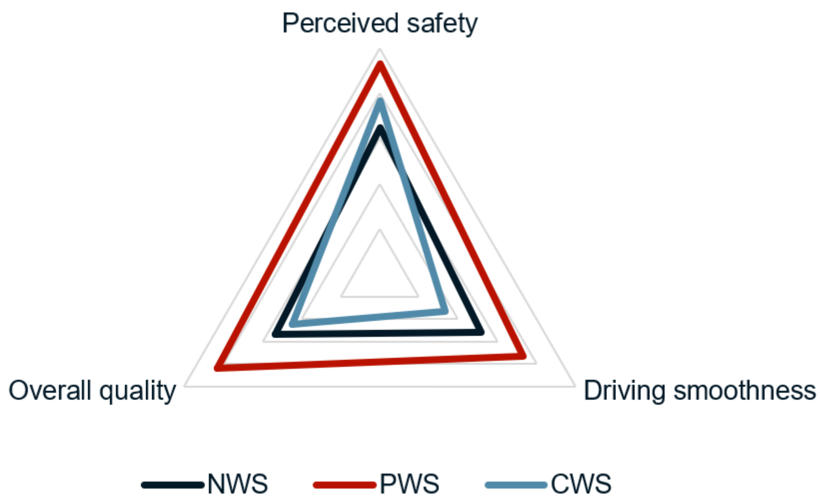


Figure 6.7: Jury's rating of driving dynamics metrics on warning systems. A larger triangular covering means better rated driving dynamics.

"Prot(ective) stop bad. Stops for 2 seconds" and" May slow-down/stop very late. Feels unsafe".

Predictive Warning System

The jury described strengths of PWS with following examples, "No protective stops but, drives relatively fast", and "Better performance while driving towards stop at wall" (refers to checkpoint 1 in segment 1), and also "Similar to NWS in effectiveness. Perceived as very safe". Four mentioned examples of weaknesses for PWS were, "Some 'jerky' behavior during driving", and "Slows down equally early for temporary objects as for permanent objects", and also "Sometimes slows down too early. Can be hard to understand why the machine is slowing down", together with "Only works on previously run routes".

Current Warning System

Some example strength on CWS were, ″Works good on straight (paths) with full speed (and) does not trigger prot(ective) stop″ and ″Easy to understand″. Example weaknesses were, ″Very jerky driving when there are obstacles off center″(refers to narrow aisle) and ″Machine drives very choppy and (has) problems with driving in all areas″.

7 Discussion

This chapter discusses the presented work in the thesis. The chapter has been divided into result discussion and method discussion. In the final section the thesis work is discussed in a wider concept.

7.1 Results

This section discusses the obtained result from Chapter 6.

7.1.1 Prediction of Position and Protective Fields

The trajectory prediction (TP) provided good prediction while driving a route that had been driven before, in most cases. As the prediction is taken from previous coordinates in the same segment it is expected to perform very well. When the CDI was facing an intersection, the TP always predicted the path which was the most common route to drive. This is a naive TP method which will predict incorrect in many cases where there are many intersections. Especially if a PWS enabled vehicle drives in intersections were different route outcomes have similar rarity. Imagine an intersection and the vehicle takes a left turn 40 percent of the time and right turn 60 percent of the time. The system will always predict driving to the right. TPs will have correct outcome 60 percent of the time and 40 percent of the time will have incorrect outcomes. This predicting method has flaws like the one just described. Another flaw is that empirical data must exist for TP to function. When an AGV is deployed in a new industry setting this method of PWS will need to drive all segments to generate an environment map. No measurement values exist for the results of TP but instead, visual results are presented in Figure 6.1. No suitable method of rating the correctness of TP was used since it is not the focus of this thesis. Therefore, it is important that known flaws, like naiveness, are discussed here. One method of measuring correctness could have been to store a predicted position ahead and compare it to the ground truth position when arriving at said position. This method was deemed not very applicable for this scenario because when driving a track without intersections the predicted coordinates exactly match the ground truth. This would have resulted in an error value of zero. When intersections were added to the track, TPs could now choose the wrong path at intersections. This would have resulted in a high error value, but the error value would be difficult to compare against anything, therefore be somewhat irrelevant. The error value would depend on how far away the paths in the intersection are to each other. This seemed like an odd way to measure strength of TP. Instead, we chose to not measure exact differences to ground truth and instead focus on discussing pros and cons with the TP method.

The future protective field prediction (FPFP) method also performed well as seen in Figure 6.2. In most turns the FPFP method chooses protective fields which matched the shape of the turn. Sometimes one field seemed out of place compared to neighboring fields and it was very hard to know if this was correct or not. One interesting point in this image is the straight field, which is the highlighted protective field in Figure 7.1, which looks out of place compared to neighboring fields but may be the exact chosen field when CDI is in this coordinate. As seen

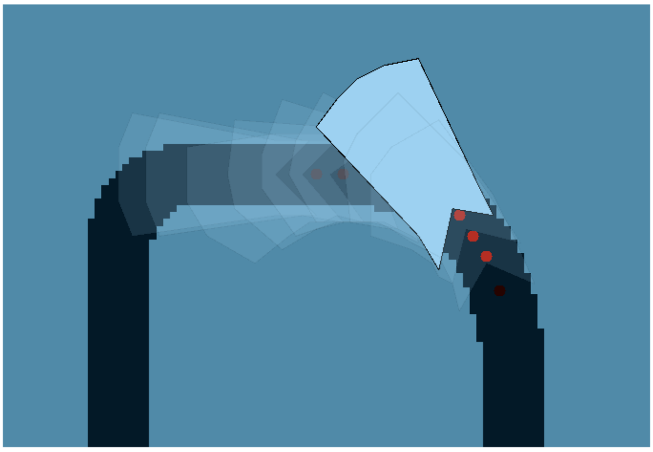


Figure 7.1: PWS estimating a protective field which appears to look off from adjacent fields. The three red dots are the given input to this choice of protective field. Note that it is the same data as 6.2, but highlighted differently.

in the figure, the field is a straight field, as if there was no turn. The three red dots are the coordinates given as input to the FPFP method. In isolation we can see that the coordinates seem to produce a straight line, which is not correct knowing that this segment is a left turn. Therefore, it looks like the FPFP method performed well given this input. The problem comes from the three coordinates that unfortunately are positioned in a straight line. If TP works as expected, these coordinates tell us that the CDI has driven past these three coordinates in this order before. This looks odd ass the curvature of this path segment should be 90 degrees and was generated from spline curve, which should not contain any straight lines. Instead, we believe the reason for why the coordinates align in a straight line comes from one of two errors. The first error is a rounding problem from discretization of coordinates to decimeter. Increasing the resolution parameter could possibly remove this error. The second error comes from observation of CDI driving on any segment. The CDI tends to slightly oscillate in position while driving on segments. Therefore, it is valid to believe that just by chance, the CDI oscillated slightly to create a straight-line coordinate pattern.

Measuring correctness of the TP method was difficult but measuring correctness for FPFP was deemed harder. Obtaining ground truth for which protective field was activated compared to our calculation would have been a very nice way to compare correctness. Sadly, no suitable method to obtain this data was achieved in the time span of this thesis work. During development, FPFP seemed like a challenging task because the method required multiple functions or modules to co-operate.

7.1.2 Driving Performance

Comparing the total time to finish test track between the warning systems relatively, we can calculate that PWS performs 17 % faster than CWS and 4 % faster than NWS. This chimes well with our hypothesis as PWS was developed to avoid the problems CWS and NWS possessed, and as the test track provoked the known problems. The exact percentages are very related to the exact shape of the test track. Therefore, it can be of interest to take a deeper look into a specific part of a segment. In Table 6.3 we get an analysis of a one-meter segment with narrow aisle. This analysis has much higher validity as the extra presented time to cover can be multiplied with the desired length in meters. Here we can clearly see the provided benefits of deploying a PWS on an AGV. As PWS does not have any *slowdown field*, speed will not rapidly increase and decrease in the same fashion as CWS. PWS will determine a valid speed to pass obstacles in future path before arriving at the obstacles.

To visualize the difference more easily in time to finish the test track for each segment Figure 7.1 was generated, taken from data in 6.5. In this figure NWS performed slowest in

Table 7.1: Different warnings systems lap time on test track for CDI vehicle.

Time to finish differences compared to NWS setup			
Warning System	Segment 1	Segment 2	Segment 3
No Warning System	0	0	0
Predictive Warning System	-3.10	0.38	-0.32
Current Warning System	-2.88	2.31	9.27

the first segment as both PWS and CWS have negative values, meaning they finished the segment faster. We can see that slightly reducing speed and taking a safer approach, like PWS and CWS for segments like segment 1 saves around three seconds driving time. It was expected that CWS and PWS would perform segment 1 faster than NWS, but it was hard to know how much faster pre-tests. It is hard to know how replicable these results are for other AGVs with similar protective fields and warning systems. Developing warning systems for AGVs at other companies may use drastically different hardware and software methods. Therefore, we would suggest readers perform their own tests on desired hardware and desired environment using our methods. This would give the reader a much more valid result. During hands on experience with the CDI and other AGVs at TMHMS we observed a non-deterministic driving behavior. With this information we would suggest taking the measured times with some prudence as the replicability may be low.

Achieving a very fast time in the NWS case is very interesting. Not having a warning system at all may give an impression of a low quality AGV, but this may not be the case. For some AGVs driving speed is of highest priority and perceived safety is not as necessary. These autonomous vehicles may never trigger protective stops operating with NWS. An example of this may be in an industrial environment without humans interfering and very few autonomous vehicles present. In cases like these, operating with NWS may be a perfectly safe and wise setup.

Segment 2 column in Table 6.1, shows us that CWS loses around two seconds to NWS in segment 2. CWS performed as expected and reduced speed for an object that was somewhat close to the path. By looking at Figure 5.5 we can see that the obstacle is very far from the path. Therefore, it was surprising to see that CWSs time to cover segment 2 was more than two seconds slower than NWS. Comparing relative time, PWS was almost 12 % faster on segment 2 than CWS. For similar environments with many turns and objects close to the path, we believe this reduced time is valid and replicable. The exact shape of the challenge of segment 2 was quite changeable. During development we observed clear difference between PWS and CWS the more objects we added close to the path, as well as the objects distance to the path impacted time to drive. More objects close to the path resulted in slower times for CWS. This shows another strength of PWS over CWS. We also believe other motion prediction models like CWS found outside of TMHMS would perform poorly in a segment like segment 2.

During driving of the test track, it could be seen that the CDI was oscillating back and forth on the right and left side of the intended drive path. This problem does not originate from any warning system or protective system, but instead comes from the on-board navigation system. The existence of these oscillations may have an impact on the warning and protective systems. Sadly, we could not prove this connection. As differential drive vehicles, like the CDI, turn by adjusting speed of each wheel, small turns will affect the relative wheel speed with large effect. Therefore, CWS may have been influenced in having an incorrect heading direction of the CDI, which in turn, reduced the speed even more than expected.

One important note on the results is that since the times are only based upon two laps it is not completely trustworthy. Since it is only two laps, the results could be based on chance instead of facts. To make it more trustworthy averaging the time over perhaps a hundred laps would be more accurate. However, this would not be possible in the scope for this thesis. Though the results are not completely significant they match the expected results and show the same behavior as during development.

7.1.3 Jury's Rating and Comments

After obtaining the knowledge of how PWS, CWS and NWS worked and seeing its ability to reduce speed for objects, the jury provided helpful insight into which warning systems they preferred. PWS topped each chart in our desired metrics for driving dynamics. It was hard to know beforehand which system the jury would favor. In Figure 6.6 we can clearly see that the jury rated PWS as both having better driving smoothness and better overall quality than CWS. This suggests that a predictive warning system is a suitable and attractive alternative to the current motion-based warning system. From the free text answers, we are given a similar story. Multiple jury members stated PWS removed protective stops while matching effectiveness level of NWS. This also gives the impression that PWS is performing well in its task. Though, the jury mentioned that PWS drove slightly too fast for their comfort. Two jury members stated that PWS had jerky driving behavior which was not wanted. One member thought the system slowed down way too early for objects in path. The members also mentioned that it can be difficult to understand why the CDI is slowing down when using PWS. When running CWS the CDI will flash shortly flash yellow to declare to its surrounding that a warning field has been triggered and it will lower speed. This is something that should have been implemented as well with PWS. All these comments provide a great imprint of what a warning system should and should not do.

7.1.4 Purpose of A Warning System and Perceived Safety

We have heard a variety of the purpose of warning systems at TMHMS. Some mentioned that its only purpose is to prevent protective and emergency stops, as these reduce the tempo and performance of the AGV. Some developers say that a warning system is essential for humans to feel safe in their environment together with the AGV. We think PWS performs well at preventing protective and emergency stops (the results also support this opinion), which in turn will increase the driving performance. Perceived safety is a hot topic in the world of autonomous vehicles and can very well be further improved with any warning system. Extensions such as light and sound when detecting nearby objects could increase the perceived safety for humans. We think all people working with AGV must be reminded that safety cannot be disabled, and protective systems are always in place. Therefore, a warning system's main purpose could be to maintain great driving performance in a very varying industry environment. Another developer explained that AGVs running with NWS perform harder braking which leads to more friction and wear on the breaks and motors. A warning system with smooth braking will reduce the wear and increase the longevity of the AGV. This developer explained a warning systems main purpose is to enable smooth driving to minimize wear. Unfortunately, no suitable method was found to measure the AGV wear from the warning systems, and therefore it is hard to determine if PWS performs well in the aspect of maintaining longevity. The jury deemed that PWS drove smoothest of all warning systems, as seen in Figure 6.6, therefore one can imagine that PWS produces the least amount of wear out of the evaluated warning systems.

7.2 Method

This section discusses the methods provided in Chapter 5 used to solve the research questions.

7.2.1 Evaluation of the Test Track

The test track is a short track, which has been developed to replicate industry environment, but may still not represent a real industry environment very well. Some drive segments for AGVs can be spanned across the whole factory and just as an example, TMHMS factory is 60000 m^2. The test track was also developed by the creators of PWS and therefore may affect PWS performance on the track as the creators may be biased towards wanting the PWS to perform well. A third-party generated test track would give each warning system a more valid and thorough performance and quality rating.

During the development of PWS, system requirements were evaluated to confirm PWS was operating as expected on multiple training tracks. These training tracks had similar segments as the test track. This resemblance indirectly means that PWS was developed to perform well on the test track. Therefore, the segments on the test track may have been an unjust demonstration in the demonstrated benefits of PWS over CWS and NWS.

7.2.2 Jury's Bias

Given the sample size of answered forms the jury ratings should be taken with prudence. Not only were there few members in the jury but the jury members might have got a subjective narrative of ”Master thesis student are replacing CWS, therefore CWS must be bad”. This narrative could have impacted PWS rating positively and CWS ratings negatively. However, one could argue that even if the jury consisted of more members it does not necessarily give more different opinions.

Another argument that could be made is the use of the 1 to 10 scale. This scale is not always the best of scales since there is no real middle point and often there will be a lot of values around five. This is because many see 10 as perfect and that is rarely the case. One thing that could have changed this argument would be if the Likert scale was used instead, that is scale where the jury would have been asked a statement and then on a scale how much they agree with the statement.

7.2.3 Improvements to the Developed Predictive Warning System

The sub-section explores methods that may improve performance of a PWS system.

Multi Outcome Prediction

TMHMS explained that an overly safe warning system that is reduces speed for too many objects outside the path is undesirable behavior. A solution that could reduce this problem, with some modifications to the PWS would be to predict every possible path. Thus, applying the protective field down each path as well. This would lead to a reduced performance of the CDI since it would slow down for objects that are not in their direct path, but instead slow down for objects in theoretical paths. What this will do on the other hand is increase the perceived safety. This is because it will slow down for everything that would be in its path.

This solution would still outperform CWS, since CWS has too wide fields which registers too many objects, and it would also have a higher perceived safety than PWS.

Improvements in Predicting Future Protective Fields

The method for predicting the protective fields could be made in different ways. One problem with the method in this thesis is that sometimes three of the points will almost be collinear and thus the code will apply a straight field. However, this is not always the case, it is possible for this to happen even in a turn. This can be seen in Figure 7.1, where in the turn a straight field is used instead of a curved one. This problem could possibly be solved by instead of using three points next to each other to create the field, three points with one point in between could be used. This would lead to bigger distances between the points and when creating the circle, it would become smaller (where the path is in a curve), thus a larger wheel velocity for the primary wheel in a turn. However, this could lead to other problems right before a turn where the path is still straight where the system might believe it would turn.

Possibly a combination of the method used in the thesis and the other method could be used, where the other method is only used in curves. This concatenation of methods could be that the skip-one method is used for the circle and if the radius is above a certain value, it will be recalculated with the method in this thesis. This would lead to a smoother behavior in curves while also behaving appropriate in straight roads.

Another method would be to reduce the number of calculative steps in the method. Some of the steps are estimating the individual wheel speeds based on radius of a turn and current speed. One method of decreasing the number of steps would be to directly read the vehicle's current wheel speeds for every coordinate and heading pair in the system and save these. With this one could use the quote of the wheels and use this to calculate the new individual wheel speeds given an average speed. Through this method, truer to life future protective fields will be estimated. This would lead to fewer errors from the FPFP method and might solve the problem shown in 7.1.

Tuning Parameters

With the method implemented in this thesis there are three parameters that can be tuned for different results. The first is at what detail the map of the world is saved at. For this thesis the world is saved at a dm level. That is, each square in the grid is 1 dm^2. If the grid had a lower resolution such a 1 m^2 it would lead to a map that would take less memory on the CDI which would be a good thing. However, what you gain in memory you lose in accuracy since it will since a lot of points will be grouped together. On the other hand, if the grid had a higher resolution, such as 1 mm^2, it would lead to better accuracy in predictions of the future path since there will be more information. The downside of this is that the memory cost would also increase drastically.

The two other parameters that could be changed are how far from the future path will be predicted as well as in how many points. In this thesis the predicted path is 8 m in 10 points. These two go very much hand in hand. If the distance is increased and the number of points remains the same, the distance, and thus the unreliability of the points, increases. The same is true if the number of points decreases while the distance stays the same. If both increase or decrease at the same rate the distance between points will stay approximately the same.

Besides that, the distance between points will decrease when adding more points and one more thing happens. Since there are more points, the time to perform the algorithm for both path prediction and laying out protective fields will increase. Thus, the number of points that can be used has a limit based on the hardware which runs the code. If the number of points would increase too far another problem would also occur. This problem is that many of the points will look to be collinear and straight protective fields will be used on almost all points. This could lead to many wrongful predictions of protective fields.

With just these three parameters there are a lot of choices to make, and in this thesis, they were just chosen at random based on what seemed to look good. This could however be investigated and an optimum, regarding to both performance but also dynamics, could possibly be found.

7.2.4 Other Predictive Methods

In this section other methods of answering the research questions will be presented.

Probabilistic Grid Method

One method that was tested early on is a method based on probability of the CDI to appear on the same place as an object. With this method, instead of predicting the protective field in different locations, the probability for the CDI to collide with the object was calculated. This was done with a similar map object as the PWS that was used. With that method only the number of times the CDI appeared in a location is saved. One can then compare the number of times the CDI has been on the object's location with the number of times it has been where it is currently. If this value is close to one the CDI should take the object seriously and slow it down. On the other hand, if it is close to zero it could take a risk and ignore it.

This method is good because it is efficient since there is no need for any larger loops or complex algorithms. However, it has some flaws, for example there is no natural speed limit to send as with PWS, and if two paths are close to each other but do not cross and something appear in the other path you might slow down for that object even though you are not on the same path.

Trajectory Predicting with Machine Learning

One way to possibly improve the trajectory prediction method would be to train a machine learning model, like a neural network or logistic regression model in estimating future position. The input base would be current position and heading and the output would be an upcoming trajectory of desired length or time horizon. It would be of great interest to test multiple extra input variables, such as current speed, load weight, vehicle position one minute ago or vehicle position ten minutes ago. The system can also be self-evaluative during run time by storing previous trajectory predictions and evaluating its correctness after passing the trajectory. Then the machine learning model could find valuable variables for trajectory prediction. With known valuable variables, trajectory prediction might give truer to life predictions, especially in junctions, compared to the PWS method described in this thesis.

However, a common problem in many machine learning solutions is solving problems with machine learning instead of using basic math and physics. A well-defined math or physics function will describe many laws of nature to greater precision than any machine learning method. Therefore, one should be restrained and careful with implementing a machine learning model for a problem like this kind.

Object Detecting and Relative Speed

A very advanced warning system would simulate detected objects path to know their future position and individual velocity. If an AGV knows where other vehicles and objects are heading, the AGV can take smart decisions to avoid unnecessary braking. To predict other objects' trajectory the AGV needs to first identify objects via the LiDAR scanner. Identifying objects with the help of 2D or 3D LiDAR is not uncharted territory [14], [11]. Today, protective systems found in literature do not take relative speed into consideration. For example, two AGVs heading straight against each other on a slightly slippery surface will most likely crash into each other with both protective system and warning system activated. Therefore, both warning systems and protective systems should take relative speed into consideration. Taking relative speed into consideration could also enable proper platooning without disabling protective fields. This is because platooning required the vehicles to drive very close to each other, but with protective system enabled, the vehicles will brake when getting close. A relative speed based protective system would notice the object in close proximity, but the system would be able to see that the upcoming vehicle is at zero relative speed. Therefore, the protective system would allow objects to drive very close without braking.

7.2.5 Reshaping Current Warning System

One could ask the question if the CWS really needs a completely new solution. It is possible to argue that it just needs some reshaping. If the *do-no-accelerate* and *slow-down fields* change shape, possibly such that the *do-no-accelerate field* inherit from the previous field, it could solve the biggest problem of the narrow aisle problem. It would solve that specific problem since the problem occurs because in some cases the *slow-down field* is larger than the *do-no-accelerate field* of one interval lower. However, this would not solve the problem in segment 2 of the test track and this problem would still exist.

7.2.6 Coordinate Discretization Influence on Predictive Warning System

One problem that exists with the current method is that when the location of the CDI is discretized there is a sharp cut-off point at .5, this is because it is rounded. This may cause a problem for the CDI when driving as the vehicle location may sporadically be rounded differently. During development, this problem could sometimes be observed when the CDI was driving on a straight segment. The problem with this is that it sometimes will predict an incorrect path. One work-around could be to borrow some logic from fuzzy logic. More specific the fact that somethings can be a bit of both and that everything does not have to be that modular. Instead of rounding the CDI could possibly start in two locations at the same time (this would not be that big of a problem since these locations would be next to each other) and then combine into the next location by averaging their individual predicted locations.

7.2.7 Computational Cost and Memory

One interesting aspect of this method is its computational and memory cost, and how it would scale when the track would increase in size. When it comes to time complexity the combine cost of the whole method would be $\approx O(n^2)$ where n is the amount of scanner beams. This shows that even if the world it operates in would get bigger it should be able to run at the same capacity. The more interesting part is its memory cost, how much memory does it need, both in the RAM, but also saved locally (this is so the CDI can save its memory of the world when it is turned off). The large culprit in the memory question is the map object which contains a lot of data. However, if it is assumed that the CDI always drives the same route and never adds any new paths then the largest part of the map will be added in the first lap of the route. Afterwards, only one number will be increased. Since these values already are integers, in code, it will take many laps until it would cause any problem and at that moment the CDI would have such a good understanding of the environment that it would be redundant to increase the counter even more. That leaves how much memory the first lap would take. With the current resolution of 1 dm^2 the map, as explained in Section 5.2, contains a total of $2 + 4 \cdot 360 = 1442$ integers (there can be a maximum of 360 headings). If that were scaled to the full size of the factory at TMHMS that would amount up to $1442 \cdot 100 \cdot 60000 = 8.652 \cdot 10^9$ integers. This is a large number of integers and is realistically not feasible and this is one of the biggest problem with this method, that it is very memory reliant. However, a small note is that the CDI would not drive everywhere but it puts into perspective how fast it grows and for this method to be applied in industry it would have to be optimized in this regard.

7.3 Thesis in a Wider Concept

When doing any form of research, it is common that the research influences other aspects, such as environmental or societal. This can either be intentional or not and this thesis is not any different. These aspects are important to look at as they can possibly show how the future would be affected by this work.

When working with automation, and specifically automating any form of driving, one important aspect is the environmental one. Currently, society is going through a automation reform which causes more electrical vehicles to be available on the market. This leads to less need for vehicles driven by fossil fuel which has a great impact on the environment. By improving the perceived safety and driving dynamics of AGVs these might have an increase in popularity and thus help society to become more automated. Since this thesis is helping with automation it is part of Industry 4.0, which is the current industrial revolution where the industry gets more automated and connected [3].

Another cause of this thesis is the computational cost. Since the method is quite heavy on the hardware it could lessen the battery life on the AGVs and thus require that the machine recharges more often. This would lead to an increase in power consumption at the company and eventually a greater impact on the power grid. On the other hand, PWS causes smoother driving which will increase the life expectancy of the machine's hardware, such as brakes, compared to CWS. The effect of this is that the hardware will require fewer changes overall and thus the production need will is also be lowered. Since the production need is going down the need for raw materials will also go down.

8 Conclusion

The research questions defined in the introduction stated the following.

1. How can a predictive warning system based on empirical data be implemented on a warehouse AGV at TMHMS?
2. How would such a warning system, compared to the current warning system at TMHMS, perform in the following aspects:
   a) *Driving performance:* Time to finish a short industry-like test track?
   b) *Driving dynamics:* Perceived safety, driving smoothness and overall quality?

Chapter 5 thoroughly explained the model architecture and steps for implementing a PWS based on empirical data which answers the first research question. The method is replicable for many other AGV systems at TMHMS but it can be difficult to know how applicable the described method is for other AGVs with vastly different architecture of on-board systems. We believe the method is applicable in other programming languages and can be implemented in systems with lower processing capabilities. Some parameters, such as resolution, may need to be lowered to not overload the system, which will impact the warning system's effectiveness. Another limitation is that the method cannot be implemented in a system without pre-defined protective fields. Another factor reducing the effectiveness of a PWS would be if the AGV very often drives on unfamiliar territory. Therefore, empirical data would not stay relevant for the system's decision making.

Regarding research questions two, our results show that a PWS based on empirical data provides slightly better driving performance over the current deployed warning system in most cases at TMHMS. In specific cases, like a narrow aisle section found in segment 3, a PWS performs much better than a motion-based warning system like the one currently deployed at TMHMS. Driving dynamics also saw better results for PWS. Firstly, perceived safety was rated highest by a PWS over both NWS and CWS. Secondly, driving smoothness, which can be described as lack of jerkiness and seamless change of velocities, was also rated highest by PWS over both NWS and CWS. Lastly, PWS got higher rating in the overall quality metric compared to NWS and CWS.

8.1 Future Works

Suggestions on improvements on the developed PWS have been provided in subsection 7.2.3 together with other predictive methods explored in 7.2.4. Future work of other warning systems evaluating performance and dynamics should take inspiration from our explored suggestions. However, some of the more interesting improvements is the use of object detection as discussed in Section 7.2.4. This method could have a major improvement on both CWS and the method used in this thesis. Here it can detect if something or someone is going towards it or just passing over the road.

Another interesting topic is to use machine learning to solve trajectory predictions. This could possibly perform very well on a path with a lot of crossings since it could find patterns such as it is doing a left turn every 10 minutes and right otherwise.

Bibliography

[1] *Argoverse 2 Motion Forecasting Dataset*. URL: https://www.argoverse.org/av2.html#forecasting-link.

[2] Timo Blender and Christian Schlegel. "Motion control for omni-drive servicerobots under Kinematic, Dynamic And Shape Constraints". In: *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*. 2015, pp. 1–8. DOI: 10.1109/ETFA.2015.7301401.

[3] Laura Boone. "Industry 4.0 (Fourth industrial revolution)." In: *Salem Press Encyclopedia* (2022).

[4] Peter Corke. *Robotics, Vision and Control - Fundamental Algorithms in MATLAB®*. 2nd ed. Springer Tracts in Advanced Robotics. Springer, 2011, pp. 110–111. ISBN: 978-3-319-54412-0.

[5] Thomas H. Cormen. *Introduction to algorithms, third edition.* MIT Press, 2009. Chap. 33. ISBN: 9780262033848.

[6] Thomas Gilles, Stefano Sabatini, Dzmitry Tsishkou, Bogdan Stanciulescu, and Fabien Moutarde. "HOME: Heatmap Output for future Motion Estimation". In: *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. 2021, pp. 500–507. DOI: 10.1109/ITSC48978.2021.9564944.

[7] Torkel Glad and Lennart Ljung. *Reglerteknik : grundläggande teori.* Studentlitteratur, 2006, pp. 191–199. ISBN: 9144022751.

[8] Kai Hormann and Alexander Agathos. "The point in polygon problem for arbitrary polygons". In: *Computational Geometry* 20.3 (2001), pp. 131–144. ISSN: 0925-7721. DOI: https://doi.org/10.1016/S0925-7721(01)00012-8.

[9] ISO Central Secretary. *Industrial trucks – Safety requirements and verification – Part 4: Driverless industrial trucks and their systems*. en. Standard ISO 3691-4:2020. Geneva, CH: International Organization for Standardization, 2020. URL: https://www.iso.org/standard/70660.html.

[10] Chiau Choon Jiat, Zarina Tukiran, Hazwaj Mohd Poad, Abd Kadir Mahamad, Sharifah Saon, Shingo Yamaguchi, and Mohd Anuaruddin Ahmadon. "Anti-Collision Car System using Fuzzy Logic Technique". In: *2021 IEEE International Conference on Consumer Electronics (ICCE)*. 2021, pp. 1–4. DOI: 10.1109/ICCE50685.2021.9427714.

[11] Daniel Maturana and Sebastian Scherer. "VoxNet: A 3D Convolutional Neural Network for real-time object recognition". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015, pp. 922–928. DOI: 10.1109/IROS.2015.7353481.

[12] Thomas K. Peucker and Nicholas Chrisman. "Cartographic Data Structures". In: *The American Cartographer* 2.1 (1975), pp. 55–69. DOI: 10.1559/152304075784447289. URL: https://doi.org/10.1559/152304075784447289.

[13] Jayanta Kumar Pothal and Dayal R. Parhi. "Navigation of multiple mobile robots in a highly clutter terrains using adaptive neuro-fuzzy inference system". In: *Robotics and Autonomous Systems* 72 (2015), pp. 48–58. ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2015.04.007. URL: https://www.sciencedirect.com/science/article/pii/S0921889015000895.

[14] Sanqing Qu, Guang Chen, Canbo Ye, Fan Lu, Fa Wang, Zhongcong Xu, and Yixin Gel. "An Efficient L-Shape Fitting Method for Vehicle Pose Detection with 2D LiDAR". In: Dec. 2018, pp. 1159–1164. DOI: 10.1109/ROBIO.2018.8665265.

[15] Jannik Quehl, Haohao Hu, Sascha Wirges, and Martin Lauer. "An Approach to Vehicle Trajectory Prediction Using Automatically Generated Traffic Maps". In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. 2018, pp. 544–549. DOI: 10.1109/IVS.2018.8500535.

[16] Matti Raitoharju, Henri Nurminen, and Robert Piché. "Kalman filter with a linear state model for PDR+ WLAN positioning and its application to assisting a particle filter". In: *EURASIP Journal on Advances in Signal Processing* 2015.1 (2015), pp. 1–13.

[17] Timothy J. Ross. *Fuzzy logic with engineering applications.* Wiley, 2017. Chap. 3-4. ISBN: 9781119235866.

[18] *Safety Systems for AGVS and AMRsSafe Efi-Pro System / microScan3 / Microscan3 Pro – Efi-Pro*. URL: https://www.sick.com/ag/en/safety-systems/safety-systems-for-agvs-and-amrs/safe-efi-pro-system/mics3-cbaz40za1p01/p/p586544.

[19] C. Schlegel. "Fast local obstacle avoidance under kinematic and dynamic constraints for a mobile robot". In: *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No.98CH36190)*. Vol. 1. 1998, 594–599 vol.1. DOI: 10.1109/IROS.1998.724683.

[20] Robin Schubert, Eric Richter, and Gerd Wanielik. "Comparison and evaluation of advanced motion models for vehicle tracking". In: *2008 11th International Conference on Information Fusion*. 2008, pp. 1–6.

[21] M. Shimrat. "Algorithm 112: Position of Point Relative to Polygon". In: *Commun. ACM* 5.8 (Aug. 1962), p. 434. ISSN: 0001-0782. DOI: 10.1145/368637.368653.

[22] Zhenglong Sun, Chuanlin Liu, and Siyuan Peng. "Maximum Correntropy with Variable Center Unscented Kalman Filter for Robust Power System State Estimation." In: *Entropy* 24.4 (2022), p. 516. ISSN: 10994300.

[23] Holger Täubig, Udo Frese, Christoph Hertzberg, Christoph Lüth, Stefan Mohr, Elena Vorobev, and Dennis Walter. "Guaranteeing functional safety: design for provability and computer-aided verification". In: *Autonomous Robots* 32.3 (Apr. 2012), pp. 303–331. ISSN: 1573-7527. DOI: 10.1007/s10514-011-9271-y. URL: https://doi.org/10.1007/s10514-011-9271-y.

[24] Vanessa Thomas. "Lidar (remote sensing technology)." In: *Salem Press Encyclopedia of Science* (2022).

[25] Varun Vaidya and Kushal Bheemesh. "Adaptive Warning Field System". MA thesis. Halmstad University, CAISR - Center for Applied Intelligent Systems Research, 2017.

[26] Abderrahim Waga, Chaymaa Lamini, Said Benhlima, and Ali Bekri. “Fuzzy logic obstacle avoidance by a NAO robot in unknown environment”. In: *2021 Fifth International Conference On Intelligent Computing in Data Sciences (ICDS)*. 2021, pp. 1–7. DOI: 10.1109/ICDS53782.2021.9626718.

[27] Kevin Weiler. “An Incremental Angle Point in Polygon Test”. In: *Graphics Gems IV*. USA: Academic Press Professional, Inc., 1994, pp. 16–23. ISBN: 0123361559.

[28] Qais Yousef, Amin Alqudah, and Shadi Alboon. “Forward vehicle collision mitigation by braking system based on artificial bee colony algorithm.” In: *Neural Computing & Applications* 27.7 (2016), pp. 1893–1905. ISSN: 09410643.

[29] Mingfeng Yuan, Jinjun Shan, and Kevin Mi. “Deep Reinforcement Learning Based Game-Theoretic Decision-Making for Autonomous Vehicles”. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 818–825. DOI: 10.1109/LRA.2021.3134249.

[30] Julius Ziegler and Christoph Stiller. “Fast collision checking for intelligent vehicle motion planning”. In: *2010 IEEE Intelligent Vehicles Symposium*. 2010, pp. 518–522. DOI: 10.1109/IVS.2010.5547976.

A Form

This chapter contains the full form for the jury to fill out during evaluation on the warning systems.

Fill this form for each warning system

1. Rate the following metrics from one to ten.
   a) Perceived Safety from ′Unsafe′ to ′Safe.′

   b) Driving Dynamics (Smoothness) from ′Clunky′ to ′Very smooth.′

   c) Overall Quality from ′Poor′ to ′High.′

2. Give examples of observed strengths in this warning system.

3. Give examples of observed weakness in this warning system.