Mälardalen University
School of Innovation Design and Engineering
Västerås, Sweden

Thesis for the Degree of Master of Science in Engineering - Robotics
30.0 credits

PATH CONTROL OF AN AUTOMATED HAULER

Fredrik Fischer
ffr11001@student.mdh.se
William Palm
wpm12001@student.mdh.se

Examiner: Giacomo Spampinato
Mälardalen University, Västerås, Sweden

Supervisor: Mikael Ekström
Mälardalen University, Västerås, Sweden

Company supervisor: Johan Sjöberg,
Volvo CE, Eskilstuna, Sweden

May 17, 2017

Abstract

The vision of self driving cars has existed for a long time and the field of autonomous vehicles has been of great interest to researchers and companies. Volvo construction equipment presented their Electrical Site project in September 2016, with predictions of reducing carbon emission up to 95% and total cost of ownership by 25%. In the project, multiple autonomous haulers are intended to work in a fleet, loading, unloading and charging in a cyclic behavior. This master thesis focus on the lateral control system of the automated hauler platform HX. The platform is modeled in an comprehensive simulation environment and three different control algorithms have been implemented and tested; An adaptive Proportional, Integral and Derivative (PID) controller, Stanley and the Proportional Integral + Proportional controller. The PID controller is tuned using the Nyquist stability criterion and the other two algorithms are tuned using a Genetic Algorithm. Results indicate that, to reach the optimal performance of the tested algorithms, manual tuning from experimental testing is required.

Table of Contents

1 Abbreviations

VTM	Volvo Trucks Model
GA	Genetic Algorithm
PID	Proportional, Integral and Derivative
PI	Proportional and Integral
MPC	Model Predictive Control
MISO	Multiple Input Single Output
IMU	Inertial Measurement Unit
EKF	Extended Kalman filter
GPS	Global Positioning System
ROS	Robot Operating System
AV	Autonomous Vehicle
COG	Center of Gravity
4WD	Four Wheel Drive
4WS	Four Wheel Steering
2WS	Two Wheel Steering
LIDAR	Light Detection and Ranging
CE	Construction Equipment
AHS	Autonomous Haulage System
D	Derivative
I	Integrating
P	Proportional
SISO	Single Input Single Output
DARPA	Defense Advanced Research Projects Agency
LAN	Local Area Network
RMS	Root Mean Square
STD	Standard Deviation
RTK	Real Time Kinematic
PATH	Partners for Advanced Transportation Technology
ADAS	Advanced Driver Assistance Systems
ITS	Intelligent Transportation Systems
V2V	Vehicle-to-Vehicle
V2I	Vehicle-to-Infrastructure

2 Introduction

This master thesis focus on the lateral control system of the automated hauler platform HX, seen in Figure 1. The main goal of the thesis is to find and implement the most suitable algorithm for controlling the hauler steering, using a Global Positioning System (GPS), Inertial Measurement Unit (IMU) and Odometry sensors, when following a specified path, at varying speed. Three different control algorithms have been chosen for implementation and evaluation. The chosen algorithms are evaluated with simulations in Simulink and experimental tests on the automated hauler platform HX.

Figure 1: Automated hauler platform, Volvo HX.

2.1 Background

The vision of self driving cars has existed for a long time and as early as in the 1950s, proportional cruise controllers were available [1]. In the 1980s, vehicle automation got a breakthrough as the microprocessor technology revolutionized the industry [1]. In the same decade, the first road following car using Light Detection and Ranging (LIDAR) and computer vision was publicly demonstrated, traveling distances up to 4.5 km at speeds up to 20 km/h along a paved road [2]. Also, a vision guided robotic van was developed by E.Dickmanns and his team [3]. The robot van could drive autonomously, on streets without traffic, with speeds up to 100 km/h. In 1994, Bundeswehr University together with Daimler-Benz presented their autonomous car *VaMP* driving in three-lane traffic with speeds up to 130 km/h. The car tracked both lane marks and other vehicles and decided when to change lanes by itself, although the approval of a human driver was required for safety reasons [4]. From this point in time, the field of Autonomous Vehicle (AV) has been of great interest to researchers and companies [5], mainly because optimal safety and comfort can be achieved [6][7] but also due to the many economical and societal benefits [8]. The field has proved great potential in many areas and applications [5]. In 2004, Defense Advanced Research Projects Agency (DARPA) launched the DARPA Grand Challenge with the goal of demonstrating the AV technical feasibility by navigating a 150-mile off-road route [9]. In the same year, the best team navigated just over seven miles and one year later five autonomous cars successfully navigated the whole route.
Today, cars with Advanced Driver Assistance Systems (ADAS) [10] as well as autopilot function-

alities such as autonomous braking, acceleration and lane guidance exist on the market, and the number of self driving functions in the cars will increase year-by-year [11].

Climate change is a grand challenge facing society and require low-carbon innovations in many sectors and industries [12], and electrified vehicles have been identified as a key technology in reducing future emissions and energy consumption within the mobility sector [13].
The benefits of electric vehicles are many as they have higher efficiency and lower operating cost compared to the conventional combustion engine vehicles [14]. Other advantages include less air pollution, less noise and vibration, lower maintenance costs and increased reliability [15].
The electrification process within the vehicle industry, has in parallel with the automation trend progressed [16] and today all major car manufacturers offer a pure battery-electric vehicle or a plug-in hybrid electric vehicle [17].

Other research conducted within the commercial vehicle industry is Intelligent Transportation Systems (ITS). ITS that utilizes Vehicle-to-Vehicle (V2V) as well as Vehicle-to-Infrastructure (V2I) communication have proven great potential as it can increase the safety, efficiency, and convenience of transportation systems [7][18][19]. In the 1990s, the California Partners for Advanced Transportation Technology (PATH) team demonstrated a platoon of 8 autonomous passenger vehicles driving on a highway with speeds up to 96km/h and a gap distance of 6.3m [20]. The V2V communication system utilized off-the-shelf wireless Local Area Network (LAN) equipment in a 900MHz band with data transmission rates of 122kbps.
Advances in communication and networking technology have made the new "cooperative vehicle paradigm" possible, but it is still unclear whether the envisioned V2V standard will become commercially available within the vehicle industry [21].

The combination of automation, electrification and connectivity of vehicles offers a variety of benefits in both performance and added values for users and businesses [22]. Connected and automated vehicles can choose routes and driving styles in order to minimize the energy consumption and ensure the best usage of the battery capacity in the electric power train for a given road profile [22]. For static environments with known road profiles such as quarries and construction sites, the energy utilization can be more easily planned and optimized. Construction vehicles usually perform repetitive tasks and cooperate with other vehicles to complete a common mission [23]. This introduces the idea of a global management system, handling the energy and productivity optimization for the entire site. The advantages and possibilities of a complete system solution have attracted the attention of the worlds leading manufacturers of construction equipment. As of the year 2015, the four largest manufacturers of construction equipment were, *Caterpillar*, *Komatsu*, *Hitachi* and *Volvo* Construction Equipment (CE) [24][25][26][27]. All of these companies have published plans on developing autonomous equipment for the commercial market.
Caterpillar announced in 2017 plans on expanding their range of autonomous trucks on the market. As results have shown that their autonomous trucks in Australia have achieved a 20 percent efficiency advantage in comparison with standard trucks [28].
Komatsu is also working on fully autonomous solutions and their comprehensive fleet management system Autonomous Haulage System (AHS), Frontrunner is already available on the market. The system has proven to contribute in reducing maintenance costs, conserving energy and curbing CO2 emissions [29].
Hitachi have, just as Komatsu, committed themselves developing an AHS. The system is scheduled to be delivered by 2017 and is expected to increase flexibility, improve safety and operation efficiency [30].
Volvo CE presented their Electrical Site project in September 2016, with predictions of reducing carbon emission up to 95% and total cost of ownership by 25%. The project aims to electrify a complete transportation solution of a quarry and involves developing new, totally electrified- as well as hybrid machines. A machine that has received a great amount of attention after the project presentation is the autonomous hauler HX, seen in Figure 1 and Figure 2. The HX hauler is in focus of this thesis and is further described in Section 2.2.

2.2 HX - Prototype Hauler

The HX is a totally electrified hauler, running only on batteries, dedicated purely for local transporting in quarries and similar transporting applications. The machine weigh 6 metric tons and is capable of transporting up to 15 metric tons of gravel.

Figure 2: Automated Hauler platform, Volvo HX.

In the Electric Site project, multiple HX are intended to work in a fleet, loading, unloading and charging in a cyclic behavior. In order for a fleet of automated haulers to work together, a well-functioning fleet management system is needed as well as stable lateral control algorithms for the vehicles. Since the choice of control algorithm has a strong impact on the performance of the vehicle, it is vital to use a controller optimized for the vehicle in order to utilize the machine to its fullest.

The main characteristics and specifications of the HX hauler are presented below:

- Mass is approximately 6 metric tons
- Loading Capacity is 15 metric tons
- Four Wheel Drive (4WD)
- Four Wheel Steering (4WS)
- Symmetric weight and form factor
- Intended working speed is 7m/s

2.3 Problem Formulation

The following problems and tasks have been identified for this thesis work.

- No accurate hauler model is available

Only a simple model, not considering tire slip etc. is available for the HX. A more complex and accurate model for the machine is necessary in order to perform valuable simulations of the control system.

- The current control system for the HX, sometimes shows smaller lateral oscillations

The control system implemented in the HX is not smooth when following a path at varying speeds. A ripple/oscillation is sometimes recorded when following a straight path.

- Current regulator does not work equally well in both forward- and reverse driving direction.

A requirement for the hauler is to be able to drive in both directions with equally good performance. The hauler is mechanically symmetric, but other parameters such as GPS position affects performance of the control system.

- When following a predefined path the hauler sometimes move outside the specified interval of $\pm n$[1].

A criteria for the control system is to be able to follow the reference path with a maximum lateral error of n. This specified criteria is not met by the current control system.

2.4 Hypothesis

Building a more complex and accurate model for the HX will allow for more accurate simulations. With simulations and tests on the real machine, a control system more suited for the specified hauler platform can be developed. The lateral control system will be smooth and stable at varying speeds and will meet Volvo CEs demands of a maximum lateral error of n.

2.5 Research Questions

- What control algorithm is most suitable for the specified hauler platform?
- Does the tested algorithms avoid oscillation when following a predefined path at varying speeds?
- Is it possible to build a simulation model that corresponds better to the HX platform?

[1]The desired maximum lateral error will be referred to as n in this thesis due to confidentiality reasons.

3 State of the Art

Control system theory has been in the scope of research since the early 20:th century [31] and today many types of controllers exists for solving control problems in most applications.
To solve the problems stated in this thesis, it is important to get a grip on control algorithms suitable for this kind of application. The algorithms in Section 3.1 - 3.9 have been used in similar applications and are thereby candidates for continued evaluation. Section 3.10 contains comparable studies between control algorithms.

3.1 PID

PID controllers are used in a wide range of industrial applications [32]. The characteristics of a PID controller includes being simple to implement and at the same time easy to tune due to reliable tuning methods such as Ziegler-Nichols method [33]. The PID controller, despite its simplicity, is a potent controller and have been used in applications where a vehicle is supposed to track a specified path [34]. A PID controller uses a feedback control loop and consists of a proportional, integrative and derivative part. The controller input is the error calculated from the measured feedback and a set-point. Controller output is calculated from a summation of the proportional error, the derivative relative to the previous iteration, and the accumulative integral error [33]. Two different forms of presenting a PID controller are commonly introduced, the *Parallel* and the *Ideal* form [35]. The *Ideal* form is the PID controller form that this thesis will use and is described by

$$\phi(t) = K_p \left[d(t) + 1/T_i \int_0^t d(\tau)\delta(\tau) + T_d \frac{\partial}{\partial x} d(t) \right] \tag{1}$$

where $d(t)$ is the lateral error, K_p is the proportional gain, T_i is the integral time and T_d is the derivative time.
The controller structure is depicted in Figure 3.

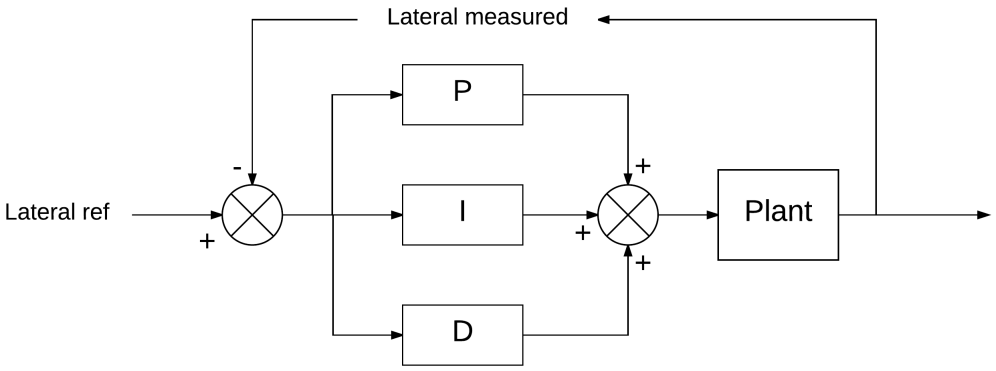

Figure 3: PID controller

Many control problems can be solved by ordinary PID controllers but depending on the control characteristics needed to solve the control problem, different compositions of Proportional (P), Integrating (I) and Derivative (D) may be preferred. The Proportional and Integral (PI) controller as an example, is one of the most commonly used control algorithms [36].
In many control applications, multiple parameters can be taken into consideration. A control system that takes multiple parameters as input, and gives a single output, is called a Multiple Input Single Output (MISO) system. A MISO system relevant in this thesis work is the PI+P control system. The PI part of the controller acts upon the lateral error and the P part acts upon the orientation error of the controlled vehicle. A more detailed explanation together with the advantages of the controller are presented in Section 8.

If the system dynamics varies depending on affecting external parameters, adaptive controllers can be favorable. An adaptive controller can change the internal gains, depending on external parameters, in order to reach stability. Adaptiveness can be added to both MISO and Single Input Single Output (SISO) control systems even though SISO control systems are sometimes preferred because of their simplicity in implementation and tuning [37].

3.2 Stanley

Gabriel M. Hoffmann et al. [38] presents the *Stanley* controlling method, developed and used by Stanford Racing Team in DARPA Grand Challenge 2005[2]. Asymptotic stability is proven for the controller using the kinematic equations of motion and the controller is extended with dynamic models for the pneumatic tires and the servo actuated steering wheel.
The complete controller solution showed great potential in both off-road and endurance tests with Root Mean Square (RMS), cross track error and Standard Deviation (STD) less than 0.08m. The team won the DARPA Grand Challenge 2005 with their new innovative controller approach, finishing the race in 6 hours and 53 minutes, giving an average velocity of 19.1 mph.
The controller on standard form can be written as

$$\phi(t) = \theta_e(t) + \arctan\left[\frac{kd(t)}{v(t)}\right] \tag{2}$$

where k is a gain variable, $v(t)$ is the velocity of the vehicle, $d(t)$ is the lateral error and $\theta_e(t)$ is the orientation error of the vehicle.
The control method uses a similar structure to the PID controller, but *Stanley* is a MISO system and uses the lateral error, orientation error and velocity as input parameters. *Stanley* uses the nonlinear feedback function arctan, for which exponential convergence can be shown [39]. The controller structure is depicted in Figure 4

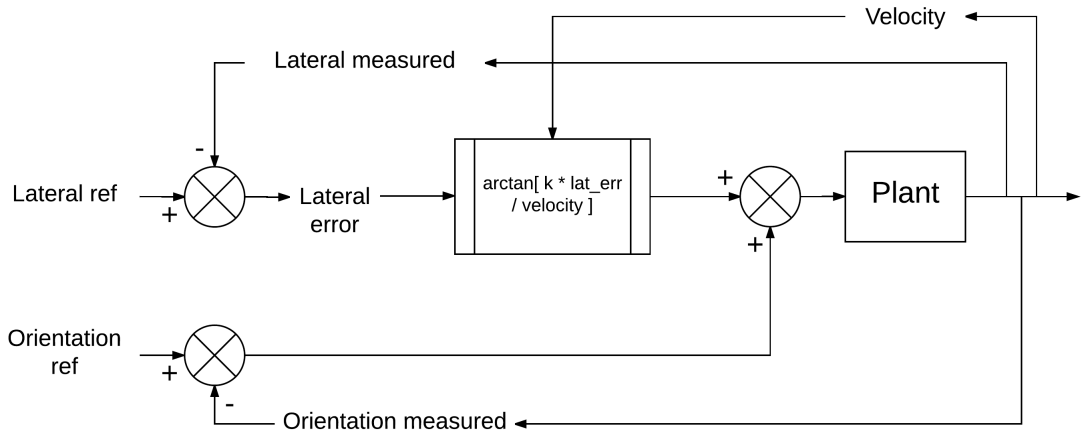

Figure 4: Stanley controller

3.3 Model Predictive Control

Model Predictive Control (MPC) is a widely used algorithm when dealing with automatic driving control [40]. The typical MPC algorithm works as described below [41][42]:

1. Measure or estimate the current system state
2. Calculate the control signal sequence

[2] A 132 miles autonomous vehicle off-road race

3. Apply the first element of the control signal sequence
4. Time update: Go to next time step and redo the process

MPC builds upon predicting the future with respect to the current state. The computational time can vary a lot depending on the complexity of the system model and the type of cost optimizing function. By measuring the system status and the desired status continuously, the controller will try to minimize the difference. This is done by taking the past control signal and system behavior into consideration while calculating future behavior using that data. An overview of the MPC concept is presented in Figure 5.

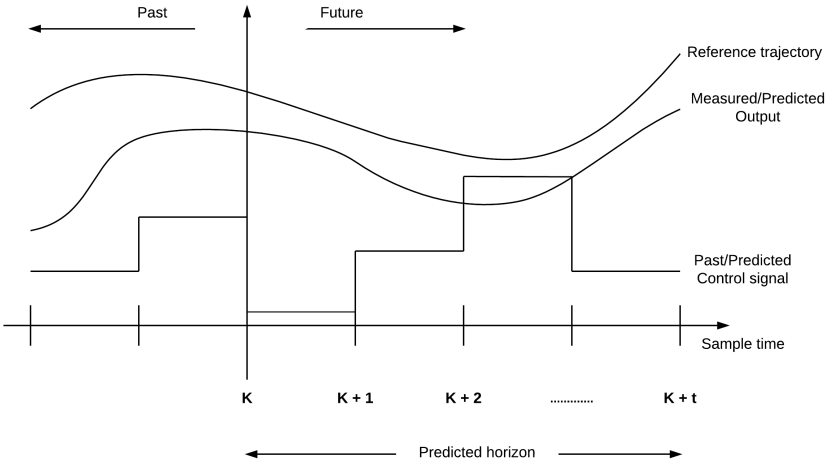

Figure 5: MPC concept overview

Wei Xi et al. [43] describes the foundation of MPC path following functionality. A scenario is described where the algorithm needs to be implemented together with obstacle avoidance. The problem is simulated using MATLAB and two methods are compared. The results are extracted by measuring the difference in computation time and error with respect to the reference trajectory. The results show that method A_1, had low computational requirements but deviated to a high degree when encountering an obstacle. Method A_2, had higher computational requirements but did not deviate as much when encountering an obstacle. Similar results have been concluded in the work of P. Falcone et al. [44] where two different MPC controllers, controller B_1 and controller B_2 were implemented and tested in a simulation environment. controller B_2 used a simpler model and a shorter prediction time, while controller B_1 used a more advance dynamic model and a longer prediction time. The goal was to track a desired path with obstacle avoidance maneuvering, using both steering and braking. The conclusions were that controller B_1 was able to stabilize the vehicle in both high and low speeds, but required a lot of computational power. While controller B_2 was not able to stabilize the vehicle at high speeds but managed to stabilize in low speeds. The article provides additional confirmation of the effectiveness of MPC but also the drawbacks of high computational requirements. MPC have also been proven to be able to handle off road conditions where sliding is a factor in the work of R.Lenain et al. [45].

3.4 Pure pursuit

Pure pursuit is a geometric control algorithm commonly used for lateral control of vehicles. The control method consists of calculating the curvature of a circular arc that connects the vehicle to a point of interest further ahead [46]. A point of interest located near the vehicle is usually referred to as a short look ahead distance. Typical behavior of a controller with a short look ahead distance is presented in Figure 6. As can be seen in the figure, the vehicle oscillates around the reference path.

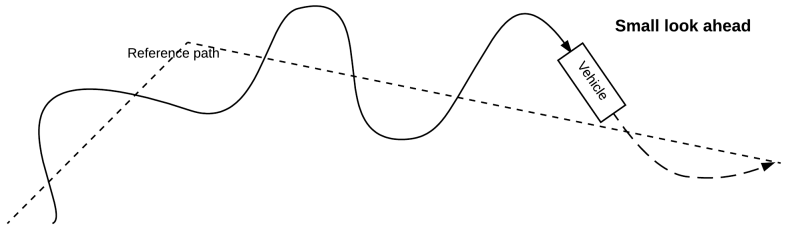

Figure 6: Pure pursuit with a short look ahead distance

The behavior of a controller with long look ahead distance is presented in Figure 7. The vehicle can seem unresponsive and tends to cut corners.

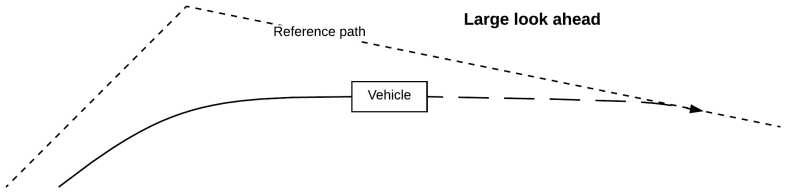

Figure 7: Pure pursuit with a short look ahead distance

When the specified road to be tracked includes multiple curves with different curvature and shape, the lookahead distance can be hard to set. Therefore many articles on automatic lookahead distance tuning have been published. Ollero et al. [47] presents a real-time fuzzy controller, automatically tuning the lookahead distance based on path characteristics, velocity, and tracking errors. Another fuzzy-supervised pure-pursuit controller for driving large autonomous vehicles at high speeds, was introduced by Rodrguez-Castano et al. [48].
Pure pursuit have been used in many types of lateral control applications due to its smooth characteristics, which is a result of using a set point in front of the vehicle. Denis Wolf et al. [49] propose a Pure pursuit control system for autonomous vehicles. The control system depends on a reduced number of parameters and is composed of a longitudinal and lateral controller. The nonlinear lateral controller utilizes a Bezier curve to find a converging path towards the planned trajectory. The vehicle follows the trajectory by manipulating the steering angle, which is computed using the current velocity and the yaw derivative of the Bezier curve. The convergence and stability of this method is studied through phase plane analysis and showed that the dynamic system converges to an attractor located in the origin, from any given state, and the error norm decreases asymptotically. The control system was tested by simulations and real tests. The results showed a good accuracy, obtaining an approximately 0m mean cross track error with a standard deviation of less than 0.1m on all of the different test tracks.

3.5 Fuzzy Control

Fuzzy control is a control system involving rule sets and membership degrees within the rule sets [50]. The rule sets are defined beforehand and decides how the system should react in different scenarios. Tuning of the fuzzy controller is performed by modifying the limits of the rules and redefining the membership functions [50]. A typical membership function setup is depicted in Figure 8.

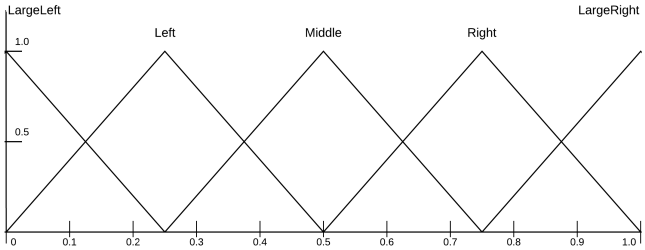

Figure 8: Fuzzy membership function

Fuzzy control have been used in lateral control applications. Xinyu Wang et al. [51] designs a steering controller using Fuzzy rule sets and membership functions, and the problems when designing a fuzzy steering controller are addressed. The presented controller considers the driving speed as the antecedent variable of each rule set, enabling forward and backward movement. Experimental results, on an electric all-terrain vehicle, showed a lateral cross track error of 1.5m from the reference path. The results of the work shows that *Fuzzy control* have limited potential for accurate path following as a standalone lateral controller.
Other applications of *Fuzzy control* have proven to be more successful. A. Visioli et al. [52] presents the benefits of using a fuzzy interface system when tuning parameters for a PID controller. The benefits are especially noticeable when dealing with nonlinear systems. Conclusions are drawn, that for some applications the same PID tuning results can be achieved with simpler methods. Though, the results show that using fuzzy when tuning PID parameters will improve performance.

3.6 H_∞

H_∞ is a control algorithm that uses an optimization function to stabilize the system. The control problem is set up as a mathematical problem, and the optimization function is then used to solve the problem [53][54]. By finding an accurate mathematical model and using a suitable optimization function, H_∞ shows great potential.
Razvan C. et al. [40] presents the possibility of using H_∞ as a lateral control algorithm. A model considering a four wheeled vehicle with front wheel steering was developed and H_∞ was implemented for lateral control. Results were evaluated by tracking the lateral error over time. The conclusions were that the control approach could provide very good performances in practical use. Computer simulations were conducted on a double lane change maneuver and a lateral error of less than 0.01m was measured.

3.7 Backstepping

Backstepping is an algorithm which uses a recursive structure to stabilize each subsystem until global stability is reached [55].
In the work of Christophe Cariou et al. [56] a 4WS vehicle was considered and a backstepping control approach was used for steering the front- and rear tires. Their main concern was the slipping in off road environments. A vehicle model was constructed and simulated, and both simulation as well as real tests were performed. The results were evaluated by tracking the lateral error over time and showed that taking slipping of the tires into consideration gave a considerably better control characteristics and a maximum of 0.05m lateral deviation from desired trajectory was achieved. To keep track of the orientation of the vehicle, a Real Time Kinematic (RTK)-GPS was used with an updating frequency of 10Hz and an accuracy of 0.02m and the vehicles orientation was extracted from a Kalman filter.

3.8 Lead-Lag compensator

Lead and lag compensator's are commonly used in different types of control systems. A lead compensator can increase the responsiveness of a system, and a lag compensator can reduce the steady state error [57]. By combining a Lead-compensator and a Lag-compensator, the result is a Lead-Lag compensator. The Lead-Lag controller removes undesired frequency response by modifying existing poles and zeros, of the transfer function, or adding additional [58]. In the work of J. Nagel et al. [57] a Lead-Lag lateral controller was implemented in the autonomous vehicle KAT-5, that competed in the DARPA grand challenge 2005. The Lead-Lag controller was based on a bicycle model and showed great results. Measurements from the initial 28 miles of the challenge showed a standard deviation of 0.05m.

3.9 Sliding mode

Sliding mode is a nonlinear control method commonly used for different control applications. A peculiar method characteristic is the discontinuous nature of the control action whose primary function is to switch between two different system structures. This behavior is claimed to result in superb system performance and insensitivity to disturbance [59].
Hiromitsu et al. [60] presents an automatic path tracking controller based on sliding control theory, for a 4WS vehicle. Path tracking for a 4WS vehicle is a more complex task in comparison with a Two Wheel Steering (2WS) vehicle, since the number of control points increases by two. In the presented controller, the front and rear wheel steering can be decoupled at the front and rear control points. Therefore, the controller can be designed more easily. Sliding mode controllers are designed and used for both the front and rear control points and the complete control system showed robustness against system uncertainties and disturbances.

3.10 Algorithms compared

Four different control algorithms used for the steering control in an autonomous car have been tested in an urban environment and evaluated by Salvador Dominguez et al. [61]. The algorithms tested were Stanley, Pure Pursuit, Sliding Mode and a Lateral speed control law. Both simulations and real tests were conducted on two different tracks. The first track was a 1km long track with a speed limit of 20km/h. The experimental results from the first track were:

- For the Pure Pursuit controller, 75% of the error was lower than 0.11m and the maximum error was approximately 0.36m.
- For the Stanley controller, 75% of the error was lower than 0.09m and the maximum error was approximately 0.4m.
- For the Kinematic sliding mode controller, 75% of the error was under 0.07m and the maximum error was approximately 0.4m.
- For the lateral speed control law, 75% of the error was lower than 0.065m and the maximum error was approximately 0.3m.

Other results showed that the algorithms possessed both negative and positive features depending on the type of application. In the comparison of the geometric controllers, Pure Pursuit and Stanley, Stanley showed slightly better results than Pure Pursuit in terms of precision.

A comparison between H_∞, Adaptive control, Fuzzy and Proportional controller was carried out by Salim Chaib et al. [62]. The algorithms were compared by simulations on a test track circuit with speeds up to 16.65m/s. It was concluded that the adaptive control, a self tuning controller proposed in the work of of R. Marino et al. [63], performs best between the compared algorithms. However R. Marino et al. do not recommend using it without considering the complexity in implementing the method. The H_∞ controller achieved equally good results as the Fuzzy controller, and the proportional controller performed the worst of the tested methods.

Additional control algorithm comparison have been conducted by Jarrod M. Snider [46]. Snider compares control algorithms used by high placed cars in the DARPA Grand Challenge, as well as other commonly used algorithms. Pure Pursuit, Stanley, Kinematic Controller, Linear Quadratic Controller with Feed Forward and Optimal preview controller were compared against each other. The algorithms were implemented and tuned before simulating the performance on three different test tracks using CarSim. The test tracks are built to test attributes of the controllers and give insight to their relative advantages and disadvantages.
The main idea of the different control methods are explained intuitively as well as methods for tuning the algorithms. The results show that the compared algorithms possessed both negative and positive features depending on the application [46] [61]. It is also concluded that some applications may benefit from a combination of approaches [46].

4 Method

This master thesis uses a qualitative methodology to analyze the chosen algorithms. The internal tuning operations follow an iterative process, as the algorithms must be rigidly tuned in order for a fair comparison to be performed. A quantitative method is used to test and verify the results by calculating statistical data for evaluation.

4.1 Method of Solution

The initial phase of the project focused on reading and researching state of the art, in order to gain a deeper knowledge on what kind of controllers that are commonly used when controlling similar vehicles in similar applications. The research showed that multiple algorithms had potential for this kind of application [46][61].
The PID controller is a favorable first approach when dealing with most control applications as described in Section 3.1. Due to the dynamic variety of the HX, as the velocity varies, a velocity adaptive PID controller was chosen as control algorithm to be implemented and evaluated.

Similar to the PID controller is the PI+P controller. The PI+P controller is a MISO control system which uses a PI part to control the lateral positioning error, and a P part adjusting the orientation error. The controllers main advantage is that the derivative part of an ordinary PID, which is the term most sensitive to noise, can be omitted. The PI+P controller is one of the algorithms that were chosen for further evaluation in this thesis work. Similar to the PID controller, the PI+P controller have to be velocity dependent due to the systems dynamic variety. A more detailed explanation together with the advantages of the controller are presented in Section 8.

An algorithm which uses a similar control system structure as both the PID and PI+P controller is the Stanley control algorithm. Stanley have turned out as a controller with great potential [38]. When comparing Stanley with other algorithms used in the DARPA Grand Challenge, Stanley showed great results [46]. Stanley is one of the algorithms that was further evaluated in this thesis work due to the performances in DARPA Grand Challenge and due to the implementation similarities to both PID and PI+P and by being velocity dependent by definition.

4.1.1 Control system tuning

A common method for tuning SISO control systems is the Nyquist stability criterion and therefore the criterion was applied to the adaptive PID controller. As a complement to the tuning method, extensive manual tuning was applied to retrieve the best controller performance possible.
PI+P and Stanley are two different types of MISO control algorithms. This introduced the problem of finding a common tuning approach. As the algorithms do not share the same characteristics, different tuning methods needs to be applied when tuning manually. Therefore, an automatic tuning approach was chosen for tuning. Using an automatic tuning algorithm makes the comparison between the algorithms more reliable as they have the same basis for achieving a satisfying result. The automatic tuning algorithm chosen is the Genetic Algorithm (GA) and is further explained in Section 6.2.

4.1.2 Simulation

A Volvo Trucks Model (VTM) modified to match the kinematics and dynamics of the HX was used to simulate the control algorithms. The simulations were performed in MATLAB 2014b and Simulink.
The first task when conducting simulations, is choosing a suitable simulation track. When running the simulations, the track should give compressive information on how the vehicle would respond, with the same settings, in real world situations.
In the work of J.M.Snider et al. [46], a track formed like the figure eight was used for simulations to measure the steady state performance and the robustness of the lateral controller.
A different approach was applied in the work of C. M. Filho et al. [49], where three different test tracks were used to get full characteristics of the tested control algorithms.

The articles indicates that it is possible to analyze the characteristics of a control algorithm using one or more test tracks. Multiple tracks take considerably longer time to simulate, due to the complexity of the simulation model and the amount of simulations that needed to be carried out. Therefore, the conclusion was to simulate the lateral control system using only one test track.
The main characteristics of the tracks mentioned in the work of both C. M. Filho et al. [49] and J.M.Snider et al. [46] were that the tracks were composed by both straight lines and sharp turns. Volvo CE's real test track is an oval track and the characteristics are suitable for this purpose. Therefore the simulation track was built to mimic the real test track, which can be seen in Figure 9.

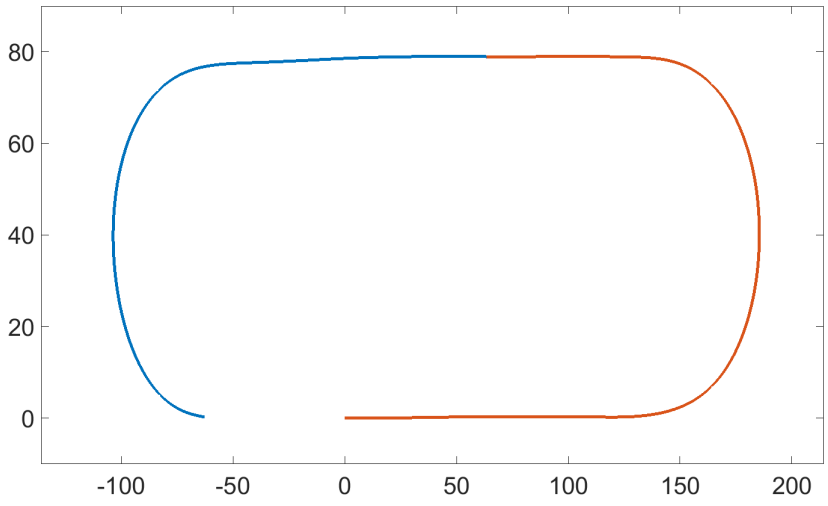

Figure 9: Overview of a the simulation track

By using only one simulation track, the simulation time to analyze the control algorithms was reduced but the overall simulation time when simulating the whole track was still regarded as too extensive. Hence, the simulation distance was limited to 300m, cutting the track distance approximately in half, see the red line in Figure 9. This resulted in a considerably shorter simulation time and consistent simulations, regardless the velocity simulated upon. Shortening the simulation distance was done without sacrificing simulation quality or outcome, as the main characteristics of the track were still retained.

4.1.3 Experimental testing

When the algorithms had been rigorously tested in the simulation environment, they were implemented in HX. This included, translating the MATLAB code to Robot Operating System (ROS) code and extracting all necessary variables from the HX main processing unit.
The tests were conducted on Volvo CEs test track for construction equipment- and prototype vehicles, seen in Figure 10. The experimental testing method used to evaluate the performance of the different controllers, are consistent with the simulation evaluation method and is presented in Section 4.1.4.

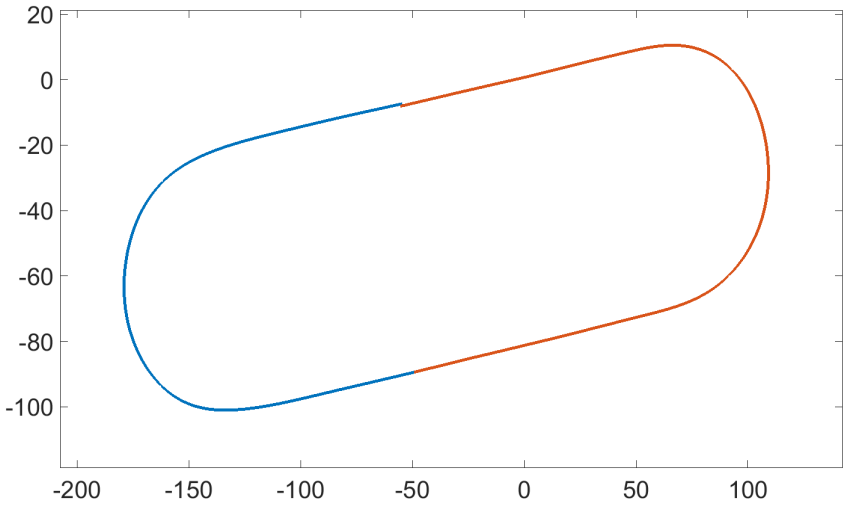

Figure 10: Overview of the test track

4.1.4 Evaluation

The method of evaluation was influenced by the work of Denis Wolf et al. [49], R. C. Rafaila [40] and C.Cariou et al. [56]. The STD of the cross track- and orientation error is calculated and taken into consideration. The fitness value calculated from the GA fitness function, described in Section 6.2.1, is also considered as an evaluation factor when comparing the different algorithms in this thesis work.
To rigidly test and evaluate the performance of the controllers, an evaluation method has been developed consisting of the following steps:

- Drive the predefined distance of 300m with a velocity of $v = 2m/s$
- Drive the predefined distance of 300m with a velocity of $v = 5m/s$
- Drive the predefined distance of 300m with a velocity of $v = 7m/s$
- Calculate STD, mean and max of the lateral- and orientation error for all of the conducted tests above
- Calculate the fitness value for all of the conducted tests above

This evaluation method was used for both simulations and experimental tests on the HX. Running the same evaluation method for both simulations and real tests will not only yield consistency, but also an understanding on how accurately the simulations correspond to reality.
The values depicted as STD, mean and max of lateral- and orientation error, presented in the result tables of each control algorithm, are normalized with respect to n and o[3] due to confidentiality reasons.

[3]The variable o was arbitrarily chosen as a normalization factor for the orientation results due to confidentiality reasons.

5 Simulation Enviroment

Due to the lack of an accurate simulation environment, a major objective in this thesis work was to modify the already existing VTM to match the HX hauler. This section will walk through some of the steps taken to modify the VTM.

5.1 Volvo Trucks Model

Volvo Group utilizes a common platform, called VTM, to simulate truck behavior in different scenarios. The platform builds upon a Simulink model consisting of multiple blocks. The blocks building the complete vehicle model are the cab, chassis, axles and tires. Each block is interconnected and is affecting the other blocks with forces. A visualization of how the blocks are interconnected can be seen in Figure 12.
The functionality and behavior of the truck can be evaluated either by using MATLAB's built in plotting tools or by using VTM Virtual Environment. The VTM Virtual Environment presents the modeled truck in a 3D environment and is presented in Figure 11.

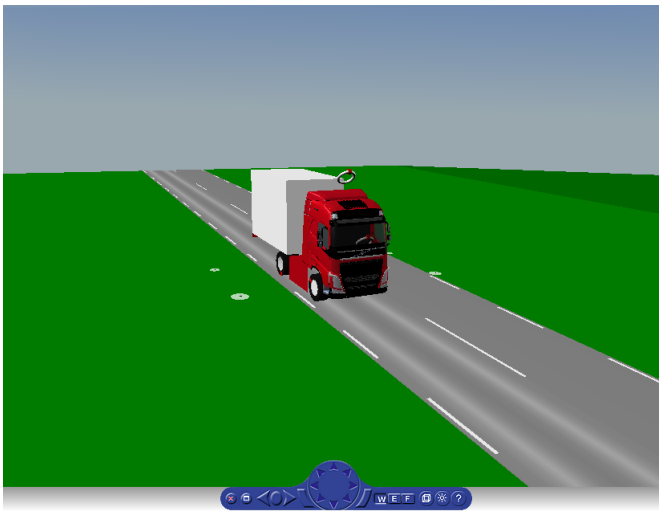

Figure 11: Volvo Truck Model - Virtual environment

As VTM provides an excellent simulation environment and modeling platform for ordinary trucks, it was decided to update and change the characteristics of an ordinary truck to match the kinematics and dynamics of the HX and utilize the advantages of the simulation environment. The major changes that were implemented includes

- Adding 4WS
- Adding 4WD
- Changing the parameters so that Center of Gravity (COG), length, weight etc. of the truck correspond to HX
- Adding steering dynamics and GPS placement

The model acting as a base for the HX model build was a 4x2 Tractor. This model was a good base for the build as it only had two axles and four wheels. The main structure of the Simulink model can be seen in Figure 12. The model is created with SimMechanics and so the information flow from the blocks flows in both directions.

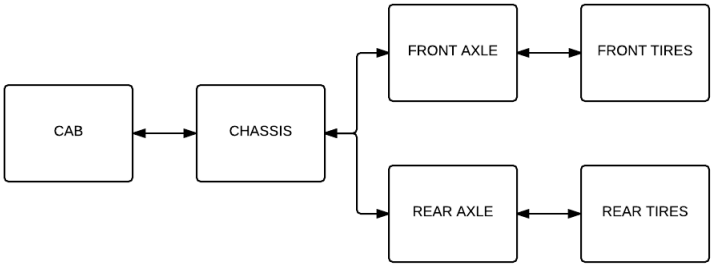

Figure 12: Simulink model - Main structure

The VTM environment does not provide a complete control system solution but only the different truck models. This meant that all the necessary parts around the plant model needed to be built. The final system architecture was built to mimic the real world application and include all the affecting factors on the lateral control system. A simple overview of the final control system structure is presented in Figure 13.
The "Controller" block contains the different lateral control systems and sends steering commands to the "Plant Model" block. The "Plant Model" block updates the state variables which are passed through a "Sensors" block back to the "Controller" block. The "Sensors" block models the different sensors, such as GPS, IMU and velocity sensor, with rate transmission and accuracy. This means that the signals from the plant model are sampled according to the matching sensor and that noise is added to the signals.

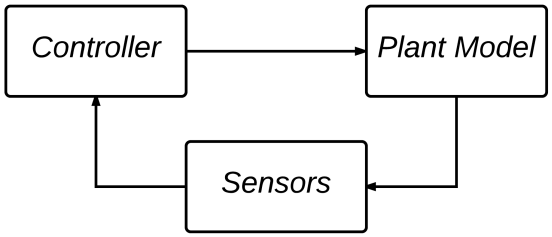

Figure 13: Control system structure

5.1.1 GPS positioning

The GPS is placed in the front right corner of the bucket on the HX. For the lateral control, the mid-point of the front axle is estimated. The position is estimated considering both the rotation of the HX body and the translation vector. As both the GPS and the IMU are not perfectly accurate and sensitive to noise, the estimated position used for the lateral control is not perfectly correct. As this can have an impact on the performance of the lateral controller, this needed to be considered in the simulations as well.

As the possibly introduced error builds upon the placement of the GPS in the XYZ-plane, and the rotation of the HX body, both the placement vector and the rotation matrix had to be considered. A vector was created consisting of the vector coordinates from the local coordinate system, located in the midpoint of the front axle, to the GPS placement on the bucket. A Body Sensor block from the SimMechanics library was then placed at the GPS vector position, taking all the rotation and warping of the bucket into account.
To mimic the GPS accuracy, the system introduces noise, in the "Sensors block", to the GPS position. This is done according to eq. (3), where GPS_{pos_est} is the estimated position of the GPS.

$$GPS_{pos\_est} = GPS_{pos} + GPS_{noise} \tag{3}$$

When the GPS position is calculated with respect to accuracy, the main objective is to track the mid-point of the HX front axle. This is done by calculating the rotation matrix of the vehicle body, then translating while rotating the inverse of the GPS vector back to the mid-point of the front axle.
The *roll*, *pitch* and *yaw* angles, needed for the rotation matrix calculation, were extracted from the vehicles Body Sensor in the plant model. To simulate the accuracy of the IMU (the sensor measuring the *roll*, *pitch* and *yaw* angles in HX), noise was added to the calculations.
The rotation matrices are defined as [64]

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\theta) & -sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix}$$

$$R_z(\gamma) = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{xyz}(\theta, \beta, \gamma) = R_x(\theta) * R_y(\beta) * R_z(\gamma) \tag{4}$$

The resulting approximation of the front axle mid-point is calculated according to

$$\begin{aligned} FrontAxle_{mid} = GPS_{pos\_est} + (-GPS_{pos} R_{xyz}(\theta + noise_\theta, \\ \beta + noise_\beta, \\ \gamma + noise_\gamma)) \end{aligned} \tag{5}$$

5.1.2 Steering dynamics

The steering dynamics is an important part of the overall system and needs to be modeled in order for the simulation model to correspond accurately to the real machine.
The steering dynamics was modeled and inserted in the Simulink model as a block inside the HX-plant. Taking demanded steering angle from the "Controller" block, processing the signal through the steering dynamics before setting the actual wheel angle.
The HX is configured to steer with symmetrical front and rear steering angle, resulting in that the vehicle will articulate around the midpoint. This was modeled by inverting the processed steering angle, before setting the actual angle of the rear wheels.

5.1.3 Additional modifications

The 4x2 Tractor model consists of one steered axle in the front and one unsteered axle in the rear. The HX is a 4WS hauler and so the rear axle needed to be replaced. As VTM is a modular simulation environment, it was easily updated with 4WS, by removing the unsteered axle and replacing it with a steered axle.

The cab in the 4x2 Tractor model was removed as HX does not have a cab inflicting dynamic effects.

Adding 4WD was done by applying torque to both the front and rear tire models.

Changing all the parameters such as weight distribution, size, wheel size, center of gravity location etc. was done by writing an initialization script for the HX model.

6 Tuning

This section will go through the tuning approaches taken to tune the different controllers. Nyquist stability criterion was applied to the PID controller as tuning approach and is further explained in Section 6.1. GA was used to automatically tune both the PI+P and Stanley algorithm and is further explained in Section 6.2.

6.1 Nyquist Stability Criterion

Nyquist stability criterion is a well know method for determining the stability of a closed loop system. By applying Cauchy's principle of argument to a open-loop system transfer function, information about the stability of the closed-loop transfer function can be received [65].
Nyquist stability criterion is only applicable to SISO systems, which the PI+P- and Stanley controller are not. Hence, only the PID controller was tuned using this approach.

In order to be able to use the Nyquist stability criterion, a linear model of the system is required. There are two different models suitable for the HX, depending on the steering configuration. The first can be described by a simple bicycle model. If the bicycle model is described around the rear wheel the model becomes [64]

$$\begin{aligned} \dot{x}_1 &= v_1 \cos(\theta_0) \\ \dot{y}_1 &= v_1 \sin(\theta_0) \\ \dot{\theta}_1 &= \frac{v_1}{L} \tan(\varphi) \end{aligned} \tag{6}$$

The second configuration is 4WS. This is the chosen configuration for the HX, mainly because it gives a symmetry to the driving.
A four wheel steered vehicle can be described by a two wheel steered bicycle model. If the model is described around the midpoint of the vehicle, the model becomes [66]:

$$\begin{aligned} \dot{x}_m &= v_m \cos(\theta_m) \\ \dot{y}_m &= v_m \sin(\theta_m) \\ \dot{\theta}_m &= \frac{1}{L}(v_0 \sin(\varphi_0) - v_1 sin(\varphi_1)) \end{aligned} \tag{7}$$

The HX is configured to steer with symmetrical front and rear steering angle, and so the vehicle will articulate around the midpoint. The previous model can therefore be simplified to:

$$\begin{aligned} \dot{x}_m &= v_m \cos(\theta_m) \\ \dot{y}_m &= v_m \sin(\theta_m) \\ \dot{\theta}_m &= \frac{2v_m}{L} \tan(\varphi) \end{aligned} \tag{8}$$

If the model is described around the front axle:

$$\dot{x}_0 = v_0 \cos(\theta_m + \varphi) \tag{9}$$

$$\dot{y}_0 = v_0 \sin(\theta_m + \varphi) \tag{10}$$

$$\dot{\theta}_m = \frac{2v_0}{L} \sin(\varphi) \tag{11}$$

The final model used for the Nyquist tuning was derived by linearizing the kinematics around a straight line and adding the steering dynamics of the system to the model.

To achieve decent robustness and stability when tuning systems, specific *undesirable regions* in the Nyquist plot have been defined, as seen in Figure 14. The gray area should be avoided and the black area must be avoided.

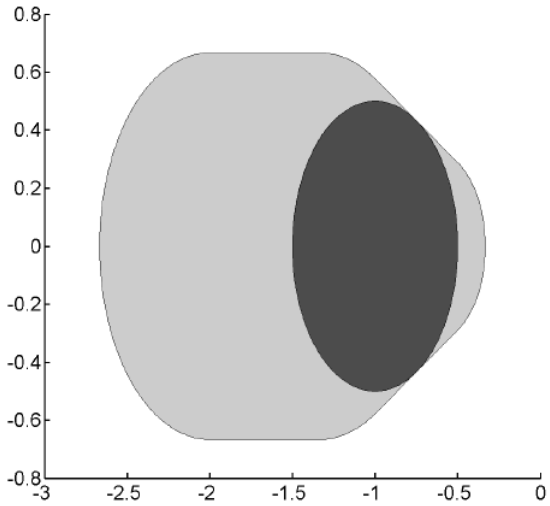

Figure 14: Undesirable regions when applying Nyquist design criteria

The idea is to tune the system parameters so that the Nyquist curve passes the undesirable regions as close as possible without crossing it. This is because the system receives a higher gain and therefore becomes more responsive closer to the regions. There is no formal motivation of the design criteria, but experience has shown that this results in a controller with a good compromise between robustness and bandwidth. More about how the undesirable regions are built, can be found in [67].
Since the system changes its characteristics drastically with speed, it is necessary to tune the system for a number of velocities. The aim is to find functions that matches the gain variations due to velocity. The velocities chosen for the tuning are

$$v = [1, 2, 3, 4, 5, 6, 7] \tag{12}$$

The Matlab built-in function *pidstd*() was used to create the pid controller defined by the different gain settings, and *nyquistplot*() was used to draw the plots to be analyzed.

6.2 Genetic Algorithm

Different types of control algorithms have been chosen for further evaluation. This introduced the problem with a consistent tuning approach. As the algorithms do not share the same characteristics, different tuning methods needs to be applied when tuning manually. Therefore, an automatic tuning algorithm was chosen as tuning approach in order to get comparable results. Multiple articles have been published on the topic of automatic tuning for control system parameters. A popular approach for tuning PID parameters is using a GA, which was presented by J. Zhang et. al. [68]. An additional approach for PID tuning is using a *Fuzzy Controller* presented by A. Visioli et al. [52]. Due to the simple implementation with MATLABs optimization package and the proved potential of GA, the GA was chosen as the automatic tuning algorithm.
GA is a well know optimization algorithm and was developed by J.H. Holland in the early 1970s [69]. The technique is inspired by nature and the natural evolution, "survival of the fittest" [70] and has received much attention in the field of locating optimal control parameters due to its high potential for solving global optimization problems [71].
A basic GA flowchart is presented in Figure 15 [72].

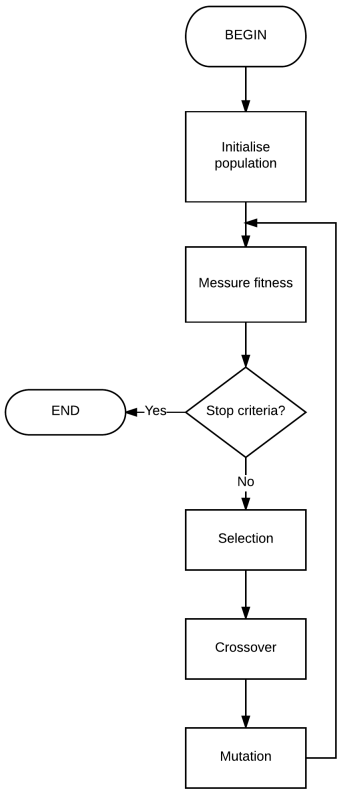

Figure 15: Flowchart of how the GA works.

The following outline summarizes the major steps in the GA [73]:

1. The algorithm begins by creating a random initial population. At each step, the algorithm uses the individuals in the current generation to create the next population.
2. Next, each member of the current population are assigned a fitness value, using the fitness function.
3. In the third step, the algorithm selects members among the population, called parents, based on their fitness.
4. Some of the individuals in the current population are chosen as elites. These elite individuals are passed to the next population.
5. The algorithm produces children, children are produced either by making random changes to a single parent or by combining a pair of parents, some of the children are mutated.
6. Replaces the current population with the children in order to form the next generation.
7. The algorithm stops when a stopping criteria, such as number of iterations, is met.

As can be seen in Figure 15, the fundamental components of the GA is the *Population*, *Fitness function*, *Selection*, *Crossover* and *Mutation*. The major operators are selection, crossover and mutation [68] and in addition there are four control parameters: population size, crossover ratio, mutation rate and in some cases the elite count [68][74]. Multiple articles suggest values for these control parameters, but the majority stay within the interval presented below [68][71][70][72].

- Population size = [50, 100]

- Crossover ratio = [60%, 80%]
- Mutation = [1%, 5%]
- Elite count = 5%

The following parameters were used in the GA optimization process of this thesis work.

- Population size = 50
- Crossover ratio = 80%
- Mutation rate = 1%
- Elite count = 5%

The values lies within the suggested interval mentioned above and are considered as default values in MATLAB documentation regarding GA [74].

6.2.1 Fitness function

The GA evaluates each individual of the population using a fitness function, and will try to minimize the cost of fitness regardless of the control algorithm it is applied to. Taking that into consideration, it is imperative that the fitness function is the same when comparing different algorithms and that it is well balanced. The main objective of the fitness function is to penalize certain characteristics of the control algorithms. The following characteristics are penalized by the fitness function defined in this thesis work:

- Large lateral error/deviation from the reference path over time
- Intense oscillations in the driven path
- Large orientation error/deviation from the reference path over time
- Large peak values of the lateral- and orientation error.

The fitness function have been constructed part by part to achieve the above mentioned functionality and the different terms of the function are presented below:

$$\omega_1 \sigma(d_{err}) \frac{1}{S_f} \tag{13}$$

The term presented in (13) will increase as the standard deviation of the lateral error d_{err} increases. A normalizing factor has been added to the term, which reduces the value to the interval of 0 to 1. The majority of the terms in the fitness function are normalized by a normalization factor.

$$\omega_2 \text{mean}\,|d_{err}|\,\frac{1}{C_f} \tag{14}$$

The term described in (14) will increase as the mean of the lateral error increases. Penalizing control tuning with high lateral error over time. The absolute value of the lateral error d_{err} is considered in this term, as the lateral error is defined as negative on one side of the reference path.

$$\omega_3 \sum \left| \frac{\mathrm{d}}{\mathrm{d}t} d_{err} \right| \frac{1}{D_f} \tag{15}$$

The term (15) will increase as the discrete-time derivative, of the lateral error d_{err}, increases. This term will penalize oscillating behavior of the controller.

$$\omega_4 \text{mean}\,|\theta_{err}|\,\frac{1}{O_{0f}} \tag{16}$$

The term described in (16) will increase as the mean orientation error increases. Penalizing controller tuning that results in large orientation error over time.

$$\omega_5 \sigma(\theta_{err}) \frac{1}{O_{1f}} \tag{17}$$

The term presented in (17) will increase as the standard deviation of the orientation error increases.

$$\omega_6 \max |\theta_{err}| \frac{1}{O_{2f}} \tag{18}$$

The term shown in (18) will penalize a high maximum orientation error.

$$\max |d_{err}| \tag{19}$$

The term presented in (19) will penalize high lateral errors. No normalizing gain have been set for this term as the maximum lateral deviation should stay below n, and if not, it is not in the scope of interest.

The final fitness function is a summation of the terms (13)-(19) and can be written as

$$\begin{aligned} Fitness = & \omega_1 \sigma(d_{err}) \frac{1}{S_f} + \omega_2 \text{mean} |d_{err}| \frac{1}{C_f} \\ & + \omega_3 \sum \left| \frac{\mathrm{d}}{\mathrm{dt}} d_{err} \right| \frac{1}{D_f} + \omega_4 \text{mean} |\theta_{err}| \frac{1}{O_{0f}} \\ & + \omega_5 \sigma(\theta_{err}) \frac{1}{O_{1f}} + \omega_6 \max |\theta_{err}| \frac{1}{O_{2f}} + \max |d_{err}| \end{aligned} \tag{20}$$

where ω determines how the following term should be weighted, d_{err} is the lateral error, σ is the standard deviation, θ_{err} is the orientation error and $S_f, C_f, D_f, O_{0f}, O_{1f}, O_{2f}$ are normalizing parameters.

7 Adaptive PID

The adaptive PID controller is an ordinary PID controller but with adaptive gains dependent on the vehicles velocity. A derivative filter was added to the controller, to reduce noise, and so the transfer function in Laplace domain becomes

$$c(s) = K_p(1 + \frac{1}{sT_i} + \frac{sT_d}{1 + \frac{sT_d}{N}}) \tag{21}$$

where the filter coefficient N was set to 10 after consulting with Volvo CEs control system expert. The controller was initially tuned using the Nyquist stability criterion but a lot of effort has been put in tuning the controller manually to maximize the controller performance. Simulations and experimental tests has been performed following the method stated in Section 4.1.4.

7.1 Implementation

The implementation of the algorithm is divided into two sections as it was implemented in Simulink/-MATLAB for simulations purposes and in ROS code on the HX for experimental testing.

7.1.1 Simulink/MATLAB

In order to evaluate the controller in the simulation environment, the controller was implemented in Simulink. The implementation was first performed by constructing a standalone block in Simulink. To ease the transfer to machine code (ROS code), it was decided to implement the controller as a MATLAB block in the Simulink environment instead.

7.1.2 Machine

In order to run the controller code on the machine, it had to be translated into ROS code and then uploaded to the main processing unit of the HX.

7.2 Simulation and Experimental testing

When the implementation in Simulink was completed the Nyquist tuning process could begin. The manually obtained tuning parameters can be seen in Table 1 and the Nyquist Plots for the different velocities in Figure 16 . The sampling time T_s for the system is 0.01s.

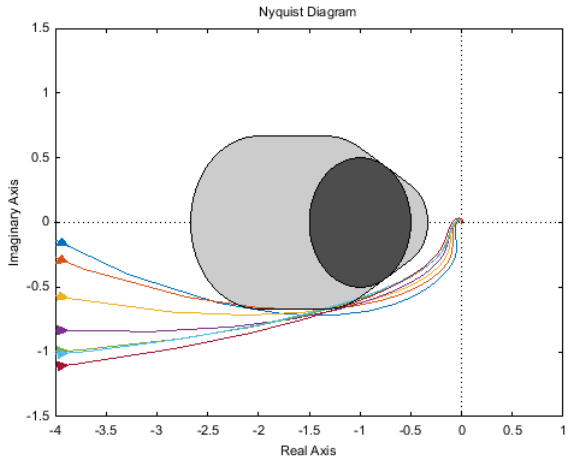

Figure 16: Nyquist plot for the system at various velocities.

v	K_p	T_i	T_d	N
1	0.55	6	0.1	10
2	0.35	5.5	0.1	10
3	0.25	7	0.1	10
4	0.2	10	0.1	10
5	0.15	20	0.1	10
6	0.11	25	0.13	10
7	0.11	26	0.13	10

Table 1: Control parameters obtained from manually tuning with Nyquist.

Plotting K_p and T_i against the velocity showed that the proportional gain was exponentially decreasing and the integral time was exponentially increasing with increasing speed. The derivative time is almost constant, as can be seen in Table 1.
The first action was to set the derivative time T_d to the constant value of 0.1 and then re-tune the system, based on the Nyquist plot. Both K_p and T_i gave the same kind of behavior as before when plotted against the velocity. The next step was to fit a polynomial, using the MATLAB built-in function $polyfit()$, for the tuned values of K_p. The polynomial defined in (22) gave an acceptable fit, see Figure 17.

$$K_p = -0.0028x^3 + 0.0488x^2 - 0.3056x + 0.8029 \tag{22}$$

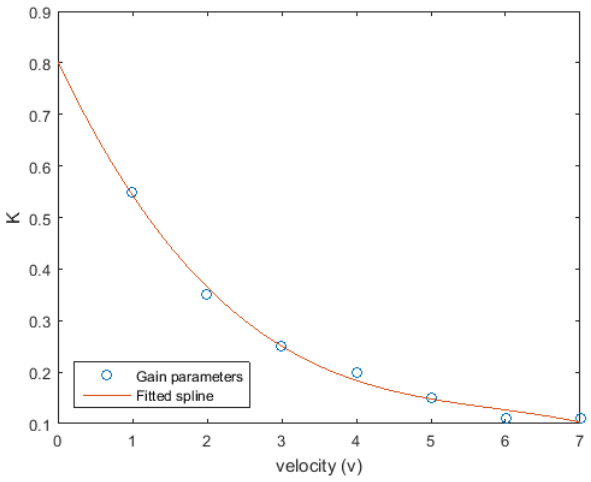

Figure 17: K_p gains plotted against velocity with fitted polynomial $K_p = -0.0028x^3 + 0.0488x^2 - 0.3056x + 0.8029$

The system was then manually re-tuned for T_i with both K_p and T_d defined and an exponentially increasing polynomial was fitted to the T_i values obtained from the tuning process. The polynomial was defined as

$$T_i = -0.1155x^4 + 1.5013x^3 - 5.3542x^2 + 6.7790x + 3.2143 \tag{23}$$

The fitted curve and the gain values can be seen in Figure 18.

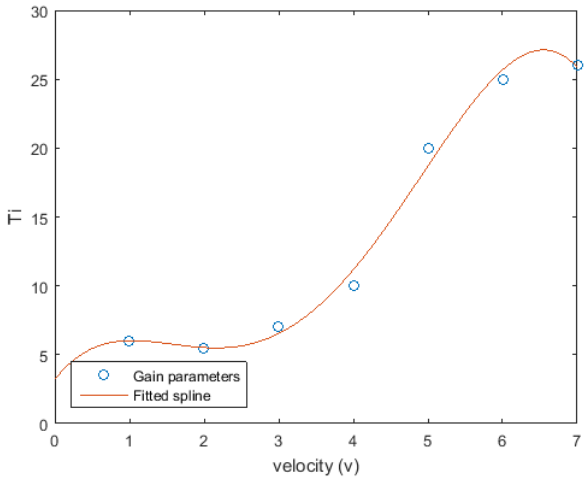

Figure 18: T_i gains plotted against velocity with fitted polynomial $T_i = -0.1155x^4 + 1.5013x^3 - 5.3542x^2 + 6.7790x + 3.2143$

Now K_p and T_i are defined as functions varying with speed. The final Nyquist plot for the control parameters can be seen in Figure 19. Only the Nyquist plot for the velocity $7m/s$ crosses the undesirable region, but not by much.

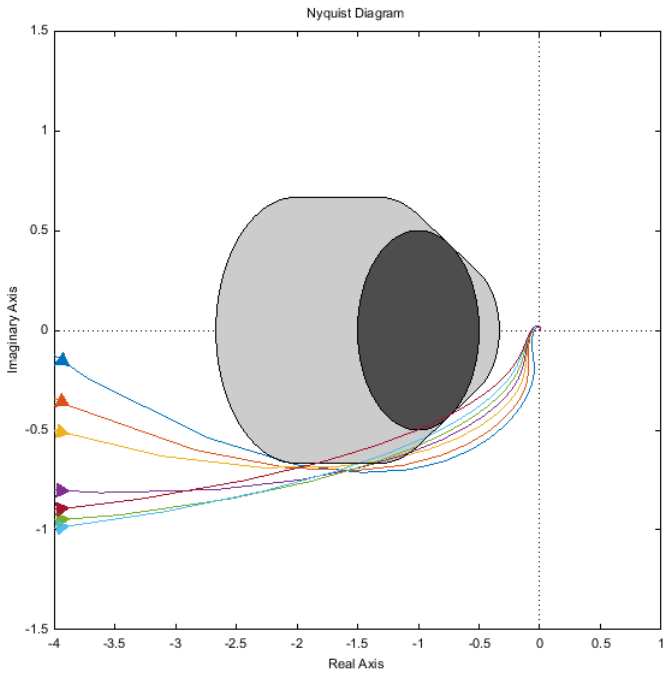

Figure 19: Final Nyquist plot for the system at various speed.

Simulating the velocity Adaptive PID did not give satisfying results as the machine oscillated a lot around the reference path. Hence, extensive manual tuning was performed, in simulations and real tests, until acceptable results were reached.

7.3 Results

The results are divided into two sections. Section 7.3.1 presents the simulation results of the controller and Section 7.3.2 presents results from experimental tests on the HX. Figures of both simulations and experimental tests are depicted in Section A.1.1 resp. A.1.2.

7.3.1 Simulation

Simulation results at speeds v= 2m/s, 5m/s and 7m/s:

Table 2: Adaptive PID Simulation results

v	STD(n)	mean(n)	max(n)	STD(o)	mean(o)	max(o)	Cost
2	0.0476	≈ 0	0.1836	16.503	8.733	57.625	1.5184
5	0.0477	0.0017	0.1967	14.417	5.69	48.079	1.2670
7	0.0677	0.0103	0.2287	13.58	3.381	41.434	1.2089

7.3.2 Experimental testing

Experimental test results at speeds v= 2m/s, 5m/s and 7m/s:

Table 3: Adaptive PID Experimental test results

v	STD(n)	mean(n)	max(n)	STD(o)	mean(o)	max(o)	Cost
2	0.3357	0.0883	1.9537	11.55	12.146	84.353	4.1929
5	0.2227	0.2773	1.0353	15.643	11.325	71.624	2.7754
7	0.4247	0.69	1.7353	11.374	9.754	91.385	3.8631

8 PI+P

The PI+P controller is a MISO system which uses a PI part to control the lateral error, and a P part adjusting on the orientation error. An explanation to the errors referred to as "orientation error" and "lateral error" are depicted in Figure 20. The lateral error is defined as the lateral difference between the middle point of the front axle and the nearest point on the reference path. The orientation error is defined as the difference between the orientation angle of the vehicle, and the orientation angle of the path at the point that the lateral error is referenced from. In Figure 20 the orientation error is depicted as Ψ and the lateral error as d.

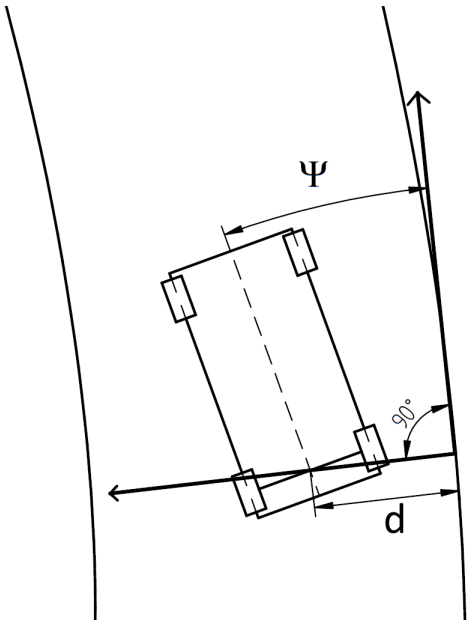

Figure 20: Explanatory figure for the terms "orientation error", Ψ and "lateral error", d.

The controllers main advantage is that the derivative part of an ordinary PID, which is the term most sensitive to noise, is not present. The PI + P controller is defined as

$$\phi(t) = P_1 d(t) + I \int_0^\tau d(t)\delta(\tau) + P_2 \theta_{err}(t) \tag{24}$$

The PI part makes the controller stable with low or no steady state error. To compensate for not using a D part in the controller, a P part was introduced to act upon the orientation error. The P part acting upon the orientation error makes it possible for the controller to be faster and more responsive.
As the path to be tracked as well as the kinematics of the vehicle is known, the optimal steering angle[4] can be calculated. The optimal steering angle is derived from the heading kinematics in eq. (11). As $\dot{\theta}_m$ is a time variable and the feed forward term needs to be dependent on distance, the following derivation was performed on eq. (11)

$$\begin{aligned} \frac{d\theta}{dt} &= \frac{ds}{dt}\frac{2}{L}\sin(\varphi) \\ \leftrightarrow \frac{d\theta}{ds} &= \frac{2}{L}\sin(\varphi) \end{aligned} \tag{25}$$

and the optimal steering angle φ becomes

$$\varphi = \arcsin(\frac{L\frac{d\theta}{ds}}{2}) \tag{26}$$

[4]The steering angle that would give perfect result if the vehicle do not deviate from the reference path.

where $\frac{d\theta}{ds}$ is the curvature of the reference path, L is the wheelbase of the HX and φ is the optimal steering angle.
φ is fed as a feed-forward term to the controller. This allows the controller to act upon the actual lateral error, rather than compensating for the curvature and the result is increased responsiveness of the controller. Figure 21 shows the complete PI + P controller structure with feed-forward as *Steering ref*.

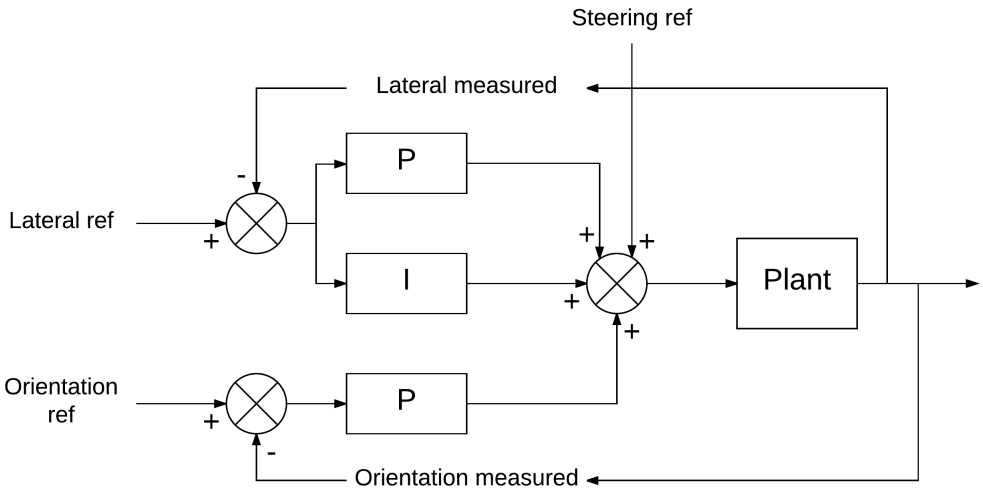

Figure 21: Overview of a PI + P regulator with feed-forward as *Steering ref*

A relationship between the derivative part of an ordinary PID controller, acting on the lateral error, and the P part of the PI+P controller that act upon the orientation error can be found. When the feedback sample time is short the relation appear more clearly. The lateral error will increase each time step if there is a steady orientation error according to Figure 22.

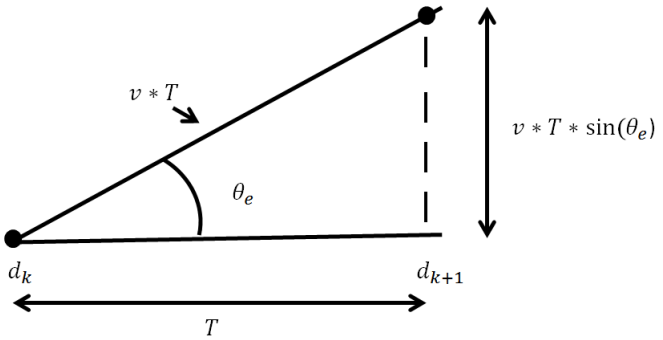

Figure 22: Relation between the orientation error and the derivative of the lateral error

where d_k is the position at time k, d_{k+1} is the position at time k+1, θ_e is the orientation error, v is the velocity of the vehicle and T is the sampling time.
When the orientation error is small, the relation can be simplified according to

$$\frac{\mathrm{d}d}{\mathrm{d}t} = v\sin(\theta_e) \approx v\theta_e \tag{27}$$

The relation can be used to run the PI+P controller on the adaptive PID controller gains, and the PI+P controller should show similar results. How the ordinary PID controller formula is translated, using the relation, to the PI+P controller formula is presented in (28).

$$c(s) = K(1 + \frac{1}{sT_i} + sT_d) = K(1 + \frac{1}{sT_i}) + KT_d v \sin(\theta_e) \approx K(1 + \frac{1}{sT_i}) + KT_d v\theta_e \tag{28}$$

One thing to comment on is the omitted filter on the derivative part of the PID transfer function, seen in eg. (28). The introduced derivative filter used in the adaptive PID will affect the results when running the PI+P controller on the adaptive PID controller gains. Setting the filter constant $N = \infty$ in eq. (21) and tuning the PID controller using the Nyquist stability criterion could be an option. But due to the fact that extensive manual tuning was needed in order for the adaptive PID to work well, extensive manual tuning would probably have to be performed after applying the Nyquist stability criterion to the PID without derivative filter. Manual tuning, from experimental testing, without a filter on the derivative part would probably be harder as the derivative part is noise sensitive and satisfactory results would probably not be achieved. It was therefore concluded to only run the relation on the already tuned adaptive PID.

8.1 Implementation

The implementation of the algorithm is divided into two sections as it was implemented in Simulink/-MATLAB for simulations purpose and in ROS code on the HX for experimental testing.

8.1.1 Simulink/MATLAB

In order to evaluate the controller in the simulation environment, PI + P was implemented in Simulink. The implementation was first performed by constructing a standalone block in Simulink. To ease the transfer to machine code (ROS code), it was decided to implement the controller as a MATLAB block in the Simulink environment instead.

8.1.2 Machine

In order to run the controller code on the machine, it had to be translated into ROS code. Also, the orientation of the machine needed to be extracted from the HX main processing unit.
In order to calculate the orientation error, the orientation of the reference path was needed. This parameter was calculated and defined as a look-up-table with respect to the path distance traveled.

8.2 Simulation and Experimental testing

The first test was to verify the relation between the derivative part of the lateral error and the orientation error, presented in Section 8. This was performed by using the already tuned adaptive PID controller gains for the PI + P algorithm when running simulations. The PI + P controller, with the adaptive PID controller gains, will hereafter be referred to as PI + P_{PID}.
The lateral error over time from simulating the adaptive PID and PI + P_{PID} controller, without any noise introduced to the orientation angle can be seen in Figure 23.

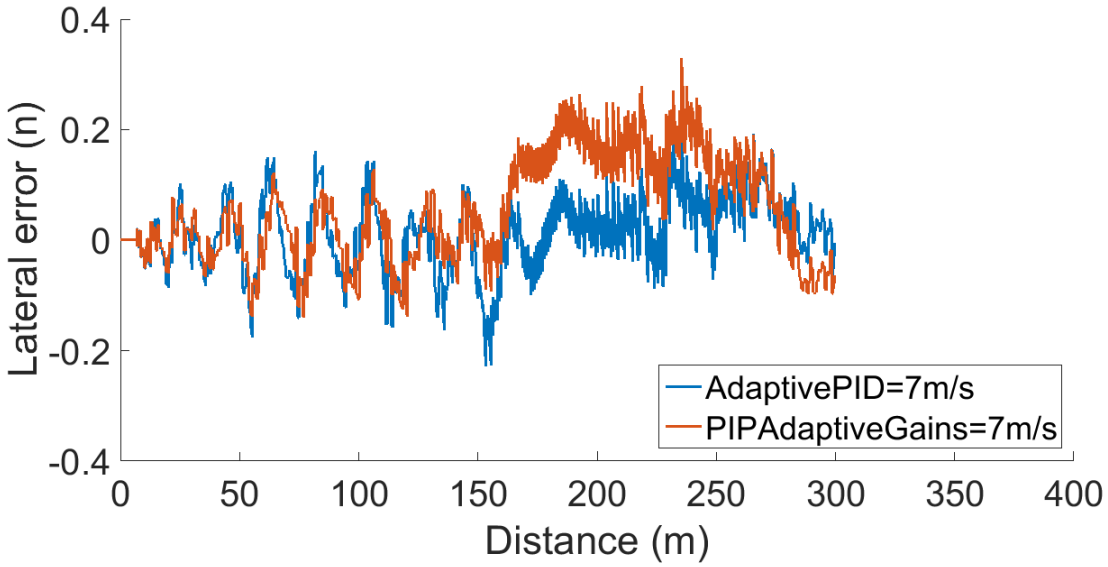

Figure 23: Lateral error over time from simulating PI + P_{PID} and adaptive PID at the speed $v = 7m/s$.

Figure 23 shows similar results and the lateral error over time shows the same characteristics in both cases. The oscillations appear to be smaller in the PI + P_{PID} controller, this could be due to the fact that the noise sensitive D part of the adaptive PID is substituted to a P part acting upon the orientation error with no introduced noise.
What should be noted is the increased lateral error as the vehicle goes into the turn of the track. This differs from the simulation results of the adaptive PID controller and could be because of the fast orientation error change that the P part of the controller is handling.
The similar response of the different controllers, and the fact that the PI + P_{PID} controller worked arbitrarily good, verifies the relation. Tests on the real machine have also been conducted, to test the relation, the results are presented in Section 8.3.2.
What can be further commented on is that the noise introduced to the orientation angle, of the machine, will affect the performance of the PI+P algorithm. Therefore, orientation noise from the existing HX model was added in the HX VTM model. Simulations results from running the PI + P_{PID} with added orientation angle noise can be seen in Section 8.3.1.

Once the relation was verified, a tuning process using GA was initialized. The parameters to be tuned were K, T_i and T_d, seen in eq. (28). The upper and lower boundaries for the tuning variables of the initial population was set to $\pm 50\%$ from the previously tuned adaptive PID gains, calculated backwards using the relation in (28). Hence, the following values were set to the initial population

$$K = \{K_{(A)} \pm 50\%\}$$
$$T_i = \{T_{i(A)} \pm 50\%\}$$
$$T_d = \{T_{d(A)} v \pm 50\%\}$$

where $K_{(A)}$ is the proportional gain from the tuned adaptive PID, $T_{i(A)}$ is the integral time from the tuned adaptive PID and $T_{d(A)}v$ is the derivative time multiplied with the velocity v assuming small θ_e according to (28).

The results from the GA optimization are presented in Table 4.

Table 4: Control parameters obtained from GA.

Velocity	K_p	T_i	T_d
1	0.5842	4.4440	0.4803
2	0.3020	12.5766	0.1364
3	0.3052	12.2455	0.2804
4	0.0188	28.4607	0.6850
5	0.0244	33.8031	2.2457
6	0.0126	32.6203	4.7308
7	0.0231	34.7578	3.9811

The variables in Table 4 were then plotted against the velocity and splines were fitted using the MATLAB built-in function $polyfit()$. The gain values with fitted splines and final adaptive functions can be seen in Figure 24 and the final functions are presented in (29) - (31). The max- and min-limits, of the defined functions, were extracted from simulation results and tests on the real machine. As too high or too low values could give unsatisfactory response from the system.

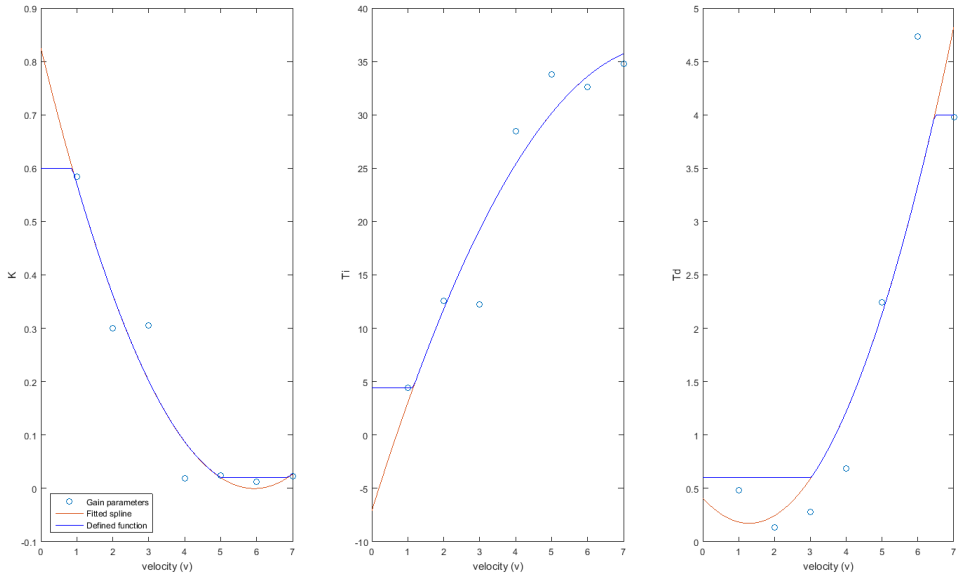

Figure 24: GA tuned gains, fitted splines and functions.

$$K = \min(0.6, \max(0.02, 0.0235v^2 - 0.2785v + 0.8257)); \tag{29}$$

$$T_i = \max(4.45, -0.6664v^2 + 10.7809v - 7.0940); \tag{30}$$

$$T_d = \min(4, \max(0.6, 0.1427v^2 - 0.3683v + 0.4103)); \tag{31}$$

As can be seen in (29) - (31) only second order splines were used to fit the tuned gain values. As for the values of T_d, seen in Figure 24, a spline with a higher degree can seem more appropriate. But simulations showed that using a higher order spline did not give sufficiently better simulation results.

Analyzing the results, from both simulations and real tests on the machine, raised suspicions that the added orientation noise affected the tuning process in an unrealistic large extent. This due to the fact that the GA tuning process was not able to retrieve tuning gains giving a sufficiently good performance.

Therefore, tests were conducted where the orientation noise was disregarded in the tuning process. The same procedure with optimization using the GA, fitting splines and building adaptive functions was executed. The resulting adaptive functions were evaluated by both simulations and experimental tests.

8.3 Results

The results are divided into two sections. Section 8.3.1 shows the statistical data collected and calculated from simulations and Section 8.3.2 shows statistical data from corresponding experimental testing. Figures depicting the lateral error over time, from both the simulations and experimental testing, can be seen in Section A.2.

8.3.1 Simulation

Statistical result data extracted from PI + P_{PID} simulations with and without orientation noise, at speeds v= 2m/s, 5m/s and 7m/s, are presented in Table 5 and Table 6 respectively.

Statistical result data extracted from PI + P_{v1} simulations with introduced orientation noise, at speeds v= 2m/s, 5m/s and 7m/s, are presented in Table 7.

Statistical result data extracted from PI + P_{v2} simulations without introduced orientation noise, at speeds v= 2m/s, 5m/s and 7m/s, are presented in Table 8.

Table 5: PI + P_{PID} simulation results (with orientation noise)

v	STD(n)	mean(n)	max(n)	STD(o)	mean(o)	max(o)	Cost
2	0.0457	≈ 0	0.1807	16.474	8.738	57.167	1.5039
5	0.1373	0.0193	0.4003	16.232	5.677	52.162	1.6822
7	0.3157	0.0487	0.8243	19.222	3.374	57.308	2.4697

Table 6: PI + P_{PID} simulation results (without orientation noise)

v	STD(n)	mean(n)	max(n)	STD(o)	mean(o)	max(o)	Cost
2	0.0457	≈ 0	0.1807	10.141	8.754	32.622	1.1120
5	0.0697	0.023	0.245	7.566	6.282	24.878	0.9381
7	0.0943	0.0547	0.3283	4.79	3.534	13.528	0.8590

Table 7: PI + P_{v1} simulation results (with orientation noise)

v	STD(n)	mean(n)	max(n)	STD(o)	mean(o)	max(o)	Cost
2	0.0478	0.0013	0.162	16.412	8.775	60.4	1.4985
5	0.1313	0.0557	0.4207	14.325	5.819	50.148	1.6111
7	0.1997	0.0787	0.519	13.847	3.467	40.093	1.6533

Table 8: PI + P_{v2} Simulation results (without orientation noise)

v	STD(n)	mean(n)	max(n)	STD(o)	mean(o)	max(o)	Cost
2	0.0377	$\approx$0	0.1527	9.919	8.789	27.049	0.9977
5	0.0593	0.017	0.1804	7.293	6.304	20.305	0.7923
7	0.087	0.036	0.2543	4.752	3.531	13.718	0.7413

8.3.2 Experimental testing

Statistical result data extracted and calculated from PI+P$_{PID}$, PI+P$_{v1}$ and PI+P$_{v2}$ experimental tests, at speeds v= 2m/s, 5m/s and 7m/s, are presented in Table 9, Table 10 and Table 11 respectively.

Table 9: PI+P Experimental testing: Using adaptive PID gains

v	STD(n)	mean(n)	max(n)	STD(o)	mean(o)	max(o)	Cost
2	0.0427	0.0243	0.3163	11.205	15.745	34.679	1.4096
5	0.0737	0.0553	0.2847	8.168	10.899	39.357	1.2525
7	0.414	0.4493	1.3233	29.195	21.315	98.566	3.8891

Table 10: PI + P$_{v1}$ Experimental testing

v	STD(n)	mean(n)	max(n)	STD(o)	mean(o)	max(o)	Cost
2	0.0423	0.0293	0.2457	10.633	15.779	37.613	1.2928
5	0.368	0.3217	1.0633	11.106	14.405	35.585	2.5296
7	0.6017	0.501	1.2367	15.519	17.849	38.963	3.0008

Table 11: PI + P$_{v2}$ Experimental testing

v	STD(n)	mean(n)	max(n)	STD(o)	mean(o)	max(o)	Cost
2	0.1257	0.0573	0.7633	14.955	14.28	95.011	2.5437
5	0.2133	0.1537	0.7263	11.826	13.781	38.296	2.0405
7	0.4227	0.3247	1.265	17.932	15.387	62.035	3.1919

9 Stanley

Stanley controller have shown great potential for lateral control of off road vehicles in rough environments. The controller uses the lateral error, orientation error and the velocity of the vehicle as input.
The basic Stanley controller has, as can be seen in (2), only one gain, k. This leaves the user with a limited possibility for adjusting and tuning. Therefore, it might not be optimal when tuning the controller for a specific vehicle, as few gains will limit the controllers possibility for optimal performance. The controller performance can be improved significantly by adding additional parameters, as it can be more application specific. Gabriel M. Hoffmann et. al. [38] proposes an extended controller formula, defined as

$$\begin{aligned}\psi(t) =&(\theta(t) - \theta_{ss}(t)) + \arctan\left[\frac{kd(t)}{k_{soft}+v(t)})\right] \\ &+k_{d,yaw}(r_{meas} - r_{traj}) \\ &+k_{d,steer}(\psi_{meas}(i) - \psi_{meas}(i+1))\end{aligned} \tag{32}$$

where k is a tuned gain and k_{soft} is a tuned gain, permitting the control to be soft at low velocities. $k_{d,yaw}$ is a tuned gain which adjusts to what extent the subtraction of the measured yaw rate of the vehicle and the trajectory yaw rate will affect the control. The purpose of this term is to create an active damping as the velocity increases. $k_{d,steer}$ is a tuned gain which adjusts to what extent previous steering angle will affect the control. This term will limit the issues in the steering, such as time delay and overshoot. θ is the orientation error and θ_{ss} is the steady state yaw. r_{meas} is the orientation of the vehicle and r_{traj} is the orientation of the nearest perpendicular trajectory. ψ_{meas} is the discrete time measurement of the steering angle and i is the index of the measurement, one control period earlier.

As can be seen in (32), a couple of terms have been added to the basic form presented in (2), the additional parameters will allow the controller to be tuned more specifically to the HX. The steady state yaw is mostly applicable for ordinary cars as they can travel in higher velocities. The steady state yaw will therefore not be used when implementing the Stanley algorithm on HX. Additionally a gain was added to the orientation error, as simulations showed that the orientation error-term became dominant in situations were small gains were needed. The controller formula that will be implemented and evaluated is described by

$$\begin{aligned}\psi(t) =&k_\theta\theta(t) + \arctan\left[\frac{kd(t)}{k_{soft}+v(t)})\right] \\ &+k_{d,yaw}(r_{meas} - r_{traj}) \\ &+k_{d,steer}(\psi_{meas}(i) - \psi_{meas}(i+1))\end{aligned} \tag{33}$$

Except from the additional terms that have been added to the controller, a feed forward term was later added to add additional responsiveness. The feed forward is calculated as described in Section 8.

9.1 Implementation

The implementation of the algorithm is divided into two sections as it was implemented in Simulink/-MATLAB for simulations purposes and in ROS code on the HX for experimental testing.

9.1.1 Simulink

Similar to the PI + P controller, the Stanley controller was first implemented using Simulink built in blocks. But to ease the transfer to machine code (ROS code), it was decided to implement the controller as a MATLAB block in the Simulink environment instead.

9.1.2 Machine

In order to run the controller code on the machine, it had to be translated into ROS code. As PI+P was implemented before Stanley, the orientation error had already been forwarded to the processing unit.

9.2 Simulation and Experimental testing

When the implementation was complete, the tuning process was initialized. Manual tuning was first used to find out in which interval the controller parameters should be located. The results of the manual tuning process is presented below

$$k = [0.1, 1]$$
$$k_\theta = [0.1, 1]$$
$$k_{soft} = [0.1, 1]$$
$$k_{d,yaw} = [0.1, 5]$$
$$k_{d,steer} = [0.1, 5]$$

The tuning process with the GA could commence, using the upper- and lower bound as an interval in which to create the initial population. The tuning resulted in the simulations depicted in Figure 34 and the corresponding results from experimental tests on the HX are depicted in Figure 36. Both simulations and experimental testing showed that the Stanley did not manage to follow the predefined path in the curve. This was found to be due to the unresponsiveness of the controller. To counter the lag, a feed forward term was added and tuning with feed forward was conducted without noise affecting the θ_{err}-term. This was due to the suspicion that the term affected the PI+P controller performance in an unrealistic large extent and would therefore affect Stanley in the same manner. The GA optimization resulted in the simulations depicted in Figure 35, and the corresponding results from experimental testing on the HX, are depicted in Figure 37.
The statistical results from both simulations and experimental tests are summarized in tables in Section 9.3.

9.3 Results

The results are divided into two sections. Section 9.3.1 shows the statistical data collected and calculated from simulations and Section 9.3.2 shows statistical data from corresponding experimental testing. Figures depicting the lateral error over time, from both the simulations and experimental testing, can be seen in Section A.3.

9.3.1 Simulation

Results of the simulation tuning process for v= 2m/s, 5m/s and 7m/s with the GA:

Table 12: $Stanley_{v1}$, conducted with noise but without feed forward

v	STD(n)	mean(n)	max(n)	STD(o)	mean(o)	max(o)	Cost
2	0.1693	-0.1497	0.5497	16.419	8.839	62.217	2.1144
5	0.445	-0.401	1.29	16.17	5.77	52.396	3.0435
7	0.6953	-0.606	1.898	20.02	3.396	63.106	4.1711

Table 13: $Stanley_{v2}$, conducted with feed forward but without noise

v	STD(n)	mean(n)	max(n)	STD(o)	mean(o)	max(o)	Cost
2	0.0583	0.0397	0.2023	9.92	8.826	27.486	1.0790
5	0.093	0.0703	0.328	7.4191	6.314	21.711	1.0380
7	0.139	0.0583	0.437	7.242	3.519	22.252	1.1696

9.3.2 Experimental testing

Using the same gains and parameters in the HX:

Table 14: $Stanley_{v1}$, corresponding tuned values from simulation 1

v	STD(n)	mean(n)	max(n)	STD(o)	mean(o)	max(o)	Cost
2	0.2267	-0.231	0.88	10.953	12.814	35.119	2.2213
5	0.56	-0.5313	1.476	17.611	14.138	98.541	3.7575
7	0.9903	-1.0353	2.6157	27.817	22.759	94.63	5.9800

Table 15: $Stanley_{v2}$, corresponding tuned values from simulation 2

v	STD(n)	mean(n)	max(n)	STD(o)	mean(o)	max(o)	Cost
2	0.0253	0.02	0.1137	10.671	11.5	32.57	0.9970
5	0.071	0.055	0.3087	9.172	9.541	34.191	1.2416
7	0.311	0.2573	0.763	17.235	14.202	51.484	2.3389

10 Discussion

The adaptive PID was initially tuned using the Nyquist stability criterion. Simulations and real tests revealed that the resulting adaptive functions, from the tuning process, did not give sufficient results and extensive manual tuning was performed. Due to the fact that the adaptive PID was tuned manually from experimental testing, a less aggressive control characteristics was achieved as can be seen in Figure 26. This is a result of performing manual tuning of the vehicle where the starting sequence is taken into consideration. A drawback is that the lateral error will take longer time converging to zero, but this kind of characteristics may be preferred in some applications in comparison to a more aggressively tuned controller.
As can be seen in Table 3, no remarkable results were achieved even though Figure 26 in some sense shows the opposite. No major oscillations can be seen and the controller slowly converges to the desired trajectory.
The simulations of the controller, seen in Figure 25, showed excellent results with a maximum lateral error of approximately $0.2n$. The authors believes that the controller could show better experimental results if the stating pose of the vehicle would be more strictly considered.

The simulation of the PI+P algorithm using the adaptive PID gains, referred to as $PI + P_{PID}$, showed good results when no orientation noise was added. Examining the simulation plots, seen in Section A, and comparing to the adaptive PID simulation plots reveals a close connection and shared characteristics. A comparison with the statistical data from the different simulations show that the PI+P algorithm has received a lower fitness cost for all of the simulated velocities even though the maximum lateral error is greater. This is because of the way the terms in the fitness function are weighted. The PI+P algorithm, in comparison with the adaptive PID, shows less frequent oscillations with smaller magnitude and this is picked up by the fitness function.
Adding orientation noise to the simulations resulted in a more unstable controller. The simulations showed greater oscillations as the velocity increased. Experimental tests on the HX, of the PI+P_{PID} controller, gave arbitrarily good results when driving at both 2m/s and 5m/s. But when reaching the speed of 7m/s, the vehicle starts to oscillate, which corresponds to the simulations with introduced orientation noise.
The $PI+P_{v1}$ controller was constructed with the adaptive PID gains as a starting point for the GA optimization. The optimization was based on the HX VTM model with added orientation noise. The resulting controller, based on velocity dependent functions, showed better simulation results in comparison with the $PI + P_{PID}$ controller with introduced orientation noise. The experimental testing did not give satisfactory results, as relatively large oscillations occurred in both 5m/s and 7 m/s. This can be due to multiple affecting factors which will be discussed further down in this section.
It was noticeable that the orientation noise had a great impact on the controller performance. Because of the fact that it was not possible to tune the controller no where near the performance of the adaptive PID, suspicions raised that the orientation noise introduced in the model was too extensive in comparison with the noise on the real machine. And that this affected the GA tuning process to not tune the controller properly. Therefore, tests without introduced orientation noise were performed. The GA tuning process retrieved completely new values and adaptive functions were made and evaluated. The simulations showed better results than the adaptive PID with a very small lateral error in all of the simulated velocities. The experimental tests gave better results than PI + P_{v1} but this could also be due to the fact that the starting pose of the vehicle was more strictly considered.
None of the tested PI+P approaches from the experimental testing showed acceptable results in terms of maximum lateral error or oscillations. All of the controllers showed similar results at a velocity of 2m/s, and experimental testing at 5m/s showed arbitrarily good results for the PI + P_{PID} and PI + P_{v2} controller. In order to build a well functioning PI+P controller, the next step would be to manually tune the controller from experimental testing. But due to lack of time, no manual tuning was performed which likely would increase the performance of the controller.

The initial tuning process of Stanley was made with introduced orientation noise and without feed forward which made the controller harder to tune. Adding a feed forward term was not planned in the initial phase of this thesis as none of the articles presenting Stanley used a feed forward term.

As can be seen in the simulation Figure 34 the controller does not manage to keep in line with the curvature. The experimental testing on the HX with the same controller parameters showed similar behavior, as can be seen in Figure 36. The similarities between experimental tests and simulations are clear when the different velocities are compared. The lateral deviation over time follow similar characteristics and receives about the same peak values as shown in the Table 12 and Table 14 in Section 9.3.
During the second tuning process, a feed forward term was added to the Stanley algorithm to improve responsiveness and to overcome the lateral deviation peak in the curve of the track. During this process, the orientation noise of the model was removed due to the suspicions mentioned earlier in this section.
The tuned parameters showed better results and the primary cause should be because of the added feed forward term, as the noise did not seem to affect Stanley in the same extent as it affected the PI+P algorithm. Simulation results are depicted in Figure 35 and the corresponding experimental tests are depicted in Figure 37.
Statistical similarities between the simulations and experimental testing are clear when comparing 2m/s and 5m/s in Table 13 and Table 15. What should be noted is that the overall characteristics of the lateral deviation differ between simulations and experimental test, which can be seen in Figure 35 and in Figure 37. This could be due to the omitted orientation noise in the simulations.

Using the GA as a tuning approach worked relatively well as the algorithm was able to achieve tuning gains that scored lower costs of the fitness function. One can question the fitness function structure and how it is weighted, but using it in the GA optimization process showed better simulation results as the function cost decreased. The weights can be tuned differently in order to receive different controller characteristics and the authors do not in any way propose that the weights used in this thesis work are optimal.
The tuning parameters retrieved from the GA optimization process worked reasonably well in the experimental testing, and the HX was able to finish the predefined test track at all the tested velocities for all tested controllers. In order to overcome the oscillation of the controllers, manual tuning from experimental testing is needed. However, the retrieved controller gains are considered as a good basis for manual tuning of the system.

The fact that the results from simulations and real tests on the HX did not exactly correspond to each other could be due to many different reasons. One aspect could be the starting pose of the vehicle. Due to difficulties in setting a perfect starting position with zero lateral- and orientation error, it will affect the tests. As can be seen in the experimental test figures, in Section A, the starting pose differs from the optimal starting pose of zero lateral error. If the starting pose lateral/orientation error is large and the controller is tuned aggressively, it will be hard for the controller to damp the oscillations once they have started.
The similarities between experimental tests and simulations with the Stanley controller indicates a close relation. The lateral deviation peaks, with approximately the same magnitude, when reaching the middle of the curve in both the simulations and real tests. The reason why Stanley shows more similar behavior between the real tests and simulations could be due to the fact that the Stanley controller is less aggressive to large lateral errors as it uses the non-linear arctan function.

Modeling the HX hauler in the simulation environment consumed more time than initially planned for in this thesis work. This resulted in less time to tune and modify the implemented control algorithms. More time needs to be put on real tests with the machine in order for the controllers to be adequately tuned.

11 Conclusion and Summary

In this master thesis work, we have modeled the HX platform in a simulation environment called VTM, the environment allows the user to build an accurate model to simulate upon. Also, a complete control system structure has been designed and implemented considering sensor accuracy and sampling limitations.
The VTM model has been used to simulate the behavior of three implemented lateral controllers, Adaptive PID, PI+P and Stanley. The adaptive PID controller has been tuned using the Nyquist stability criterion and extensive manual tuning have been performed. The PI+P and Stanley controller have been automatically tuned using a GA approach.
Simulations and experimental tests have been performed and the control algorithms have been evaluated. All of the algorithms have proven potential for this application and the following conclusions can be drawn from this thesis work.

- The built VTM model gives a good indication on how the control system will perform on the real machine. However, manual tuning from experimental tests is needed to obtain the controllers optimal performance.
- The adaptive PID controller performed best among the tested controllers, when considering the experimental test performance. This is due to the defensive controller characteristics, which is better suited for this application.
- Both PI+P and Stanley show an oscillating behavior, but the adaptive PID does not show any signs heavy oscillations when following a predefined path.

12 Future Work

During this thesis work, some obstacles was encountered that was not included in the initial time plan, making some of the initial goals unreachable within time limit that this thesis covered. To update the VTM model to match the HX consumed more time than initially planned, which limited the work to cover three algorithms instead of four. Additionally there was no time to try the implemented algorithms in both backward- and forward driving direction, as well as in loaded- and unloaded state. Only an unloaded HX with forward driving direction was covered in this thesis.

The next step for the tested control algorithms would be to measure the real orientation accuracy on the HX platform. Since the introduced orientation noise in the model has such a big impact on the control algorithms performance, especially on the PI+P controller, it is vital to simulate with the correct level of noise.
A lot of effort need to be put on manually tuning the algorithms with experimental tests on the HX. The model, that the tuning has been performed against, cannot correspond exactly with the HX platform. This implies that manual tuning might be the best way to make the final tuning adjustments to the control algorithms, in order to receive the best possible outcome.
The algorithms in Section 12.1 and Section 12.2 are control algorithms that has been considered to have great potential for this kind of application. Implementing and evaluating these algorithms might be beneficial as a future work.

12.1 Model Predictive Control

MPC utilizes a model to predict how the future control signal will affect the system and a cost function intended to be minimized. Usually the function is defined as the distance between the reference path and the current location, for this kind of application [42]. The model is usually simplified to allow the controller to be faster, as computational power is often an issue in embedded systems. A vehicle in motion can be described by [42]

$$\dot{x} = Ax + B_1 u + B_2 w + B_3 r \tag{34}$$

$$y = Cx + Du \tag{35}$$

$$\dot{x} = \text{Change in the vehicle state}$$
$$y = \text{Manipulation vector}$$
$$u = \{\phi\} = \text{Steering control vector}$$
$$x = \left\{v, XY_{pos}, \dot{\phi}, \phi, \dot{\theta}, \theta\right\} = \text{State vector}$$
$$r = \{XY_{des}, \phi_{des}\} = \text{Desired state vector}$$
$$w = \{Noise, Delay\} = \text{Disturbance vector}$$

Given the system vectors and the objective of the path controller, a cost function can be formulated. The proposed cost function is displayed in eq. (36).

$$Cost(k) = \sum_{i=1}^{H_p} |y(k+i|k) - r(k+i|k)|^2 + \sum_{i=1}^{H_c-1} |\Delta u(k+i|k) - r(k+i|k)|^2 \tag{36}$$

$$i = \text{Sample time}$$

$$k = \text{Current sample time}$$

$$H_p = \text{Predictive control horizon}$$

$$H_c = \text{Control signal sequence}$$

Where the first summation aims to minimize the difference between the reference trajectory and the current position. The second summation will limit the controller output, which will limit the controller to max the steering. The main benefit with the MPC controller, is the ability to continuously adapt the controller according to the reference model. This allows the model to take noise and delays of the system into consideration. Therefore, the MPC controller would presumably react better to disturbances in comparison to controllers not considering these affecting factors.
The authors of this thesis believe that, because of the many advantages of MPC in comparison with more simple controllers, an implementation of MPC as future work could be beneficial and give a satisfactory results.

12.2 Pure Pursuit

One of the most common approaches when dealing with path following for mobile robots, is pure pursuit [46]. The implementation is considered as fairly simple and all the necessary parameters are already available in the processing unit of the HX.
The pure pursuit controller law is shown in eq. (37).

$$\phi(t) = \tan^{-1}\left(\frac{2L\sin(\alpha(t))}{l_d}\right) \tag{37}$$

$$\phi(t) = \text{Steering angle}$$

$$\alpha(t) = \text{Vehicle orientation error}$$

$$l_d = \text{Look ahead distance}$$

$$L = \text{Length between the wheel axles}$$

The tuning is usually done by adjusting the look ahead distance dependent on the vehicle speed multiplied with a tuned gain [46]. An alternative could be to adjust the look ahead distance depending on the curvature of the reference path, as pure pursuit may be cutting corners with a long look ahead distance and large curvatures. Because of the simplicity of implementation, the algorithm should be considered as future work as it could give interesting results.

13 Acknowledgement

We would like to express our gratitude to our supervisors Johan Sjöberg and Mikael Ekström for their great support throughout the project. A special thanks to Ted Samuelsson and Markus Rombach who invested a lot of time and patience during tests and result acquisition.

A Results

A.1 Adaptive PID

A.1.1 Simulations

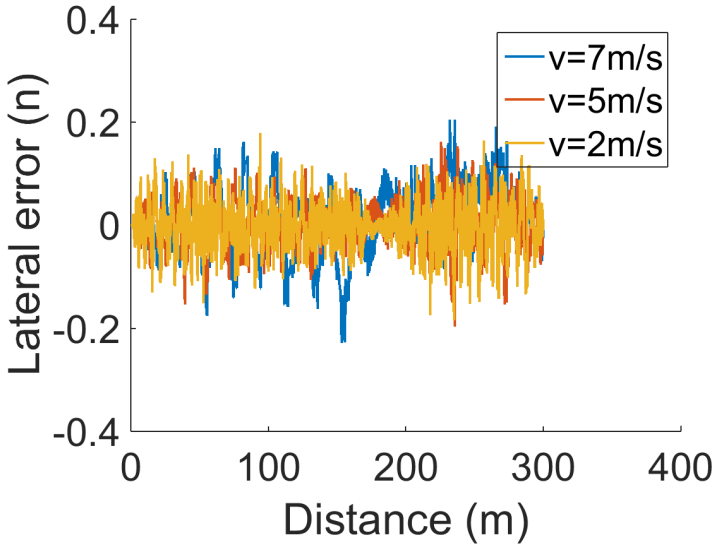

Figure 25: Adaptive PID in simulation environment

A.1.2 Experimental testing

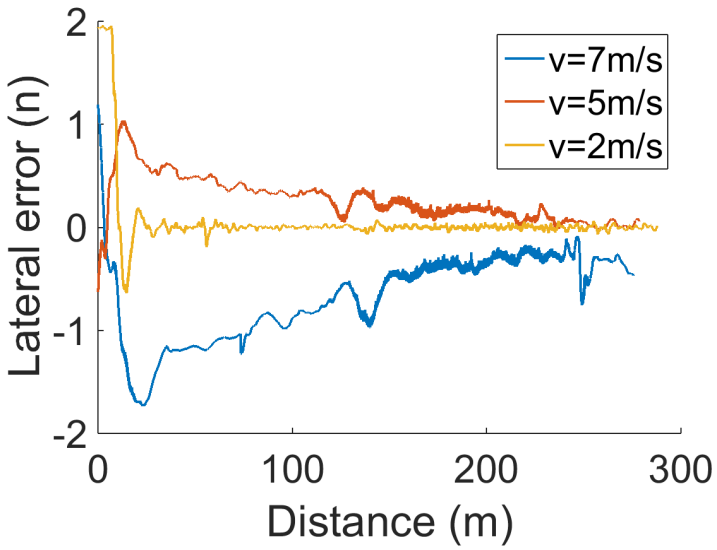

Figure 26: Adaptive PID on test track

A.2 PI+P

A.2.1 Simulations

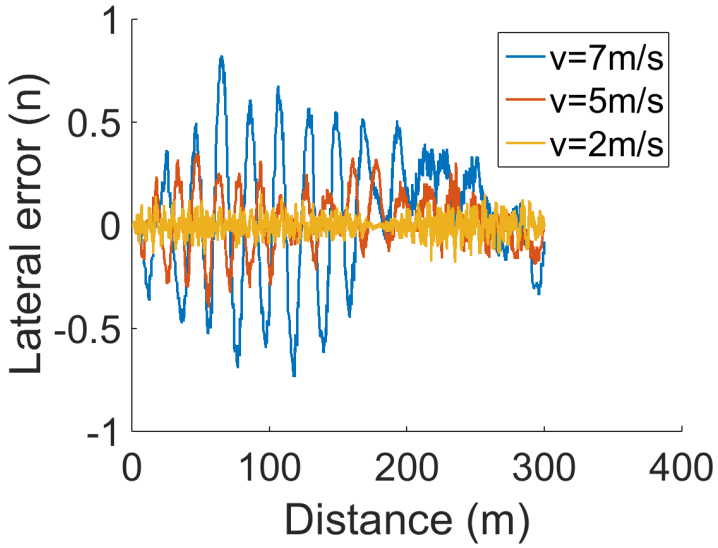

Figure 27: $PI + P_{PID}$ in simulation environment with added orientation noise

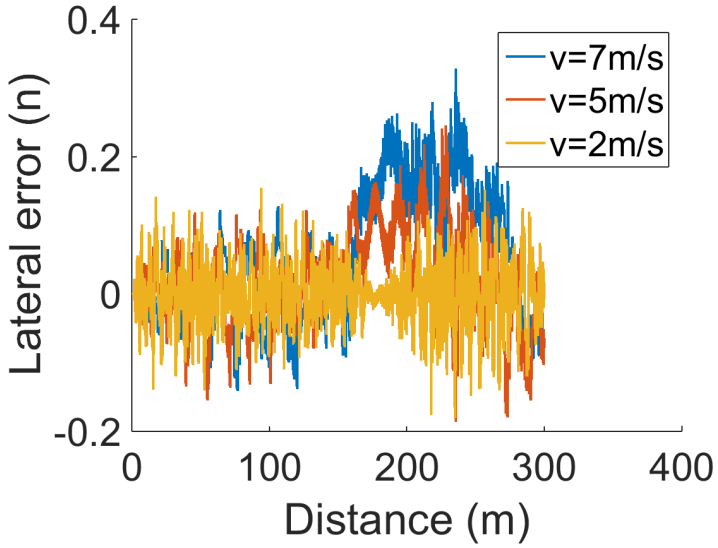

Figure 28: $PI + P_{PID}$ in simulation environment without orientation noise

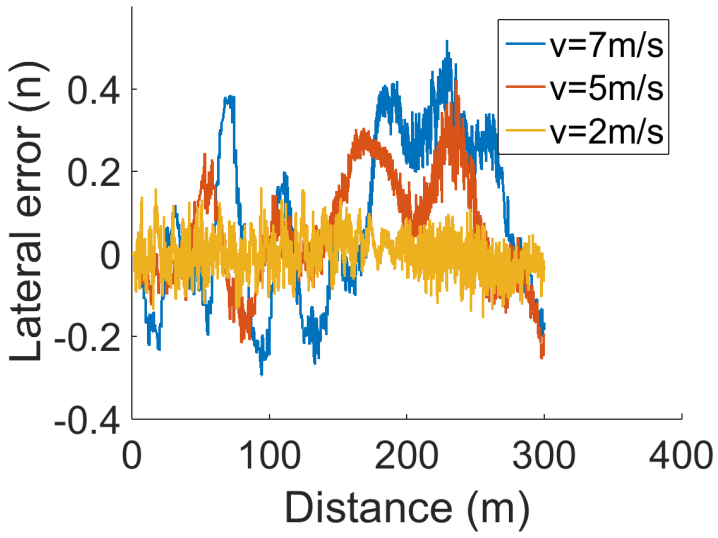

Figure 29: $PI + P_{v1}$ in simulation environment with added orientation noise

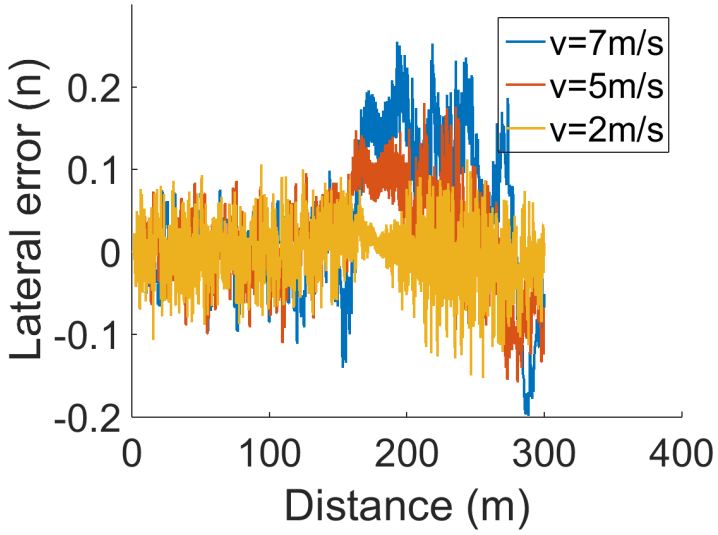

Figure 30: $PI + P_{v2}$ in simulation environment without orientation noise

A.2.2 Experimental testing

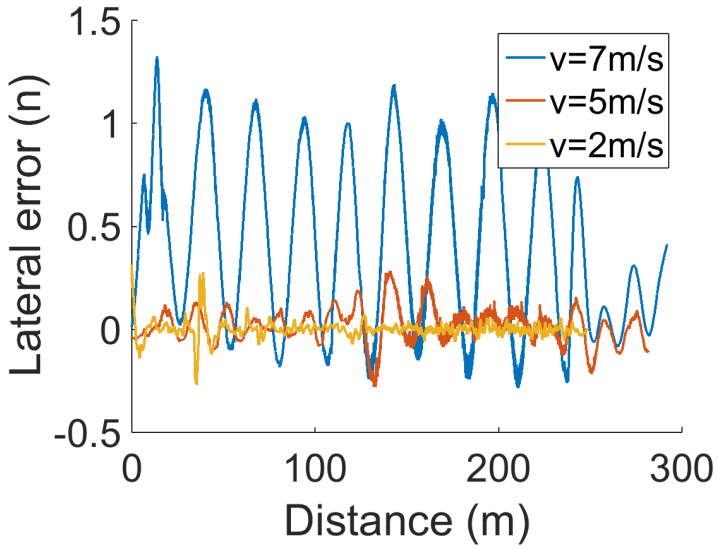

Figure 31: PI + P Experimental Testing: Lateral error over time using Adaptive PID gains

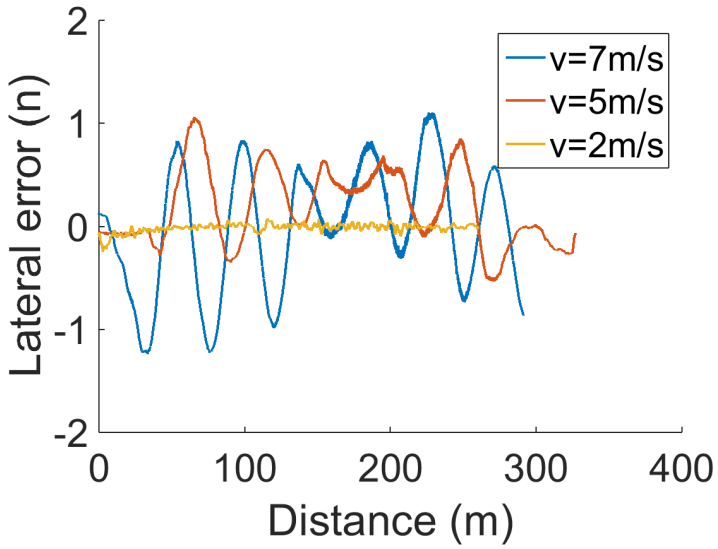

Figure 32: PIP_{v1} Experimental Testing: Lateral error over time with orientational noise

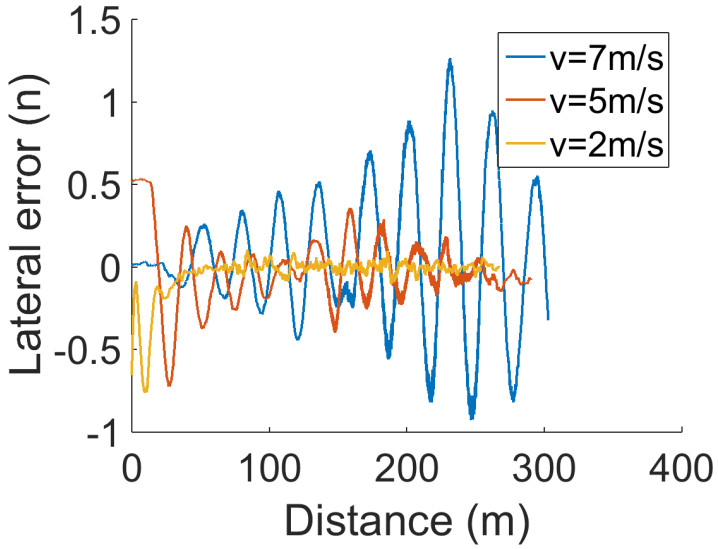

Figure 33: PIP_{v2} Experimental Testing: Lateral error over time without orientational noise

A.3 Stanley

A.3.1 Simulations

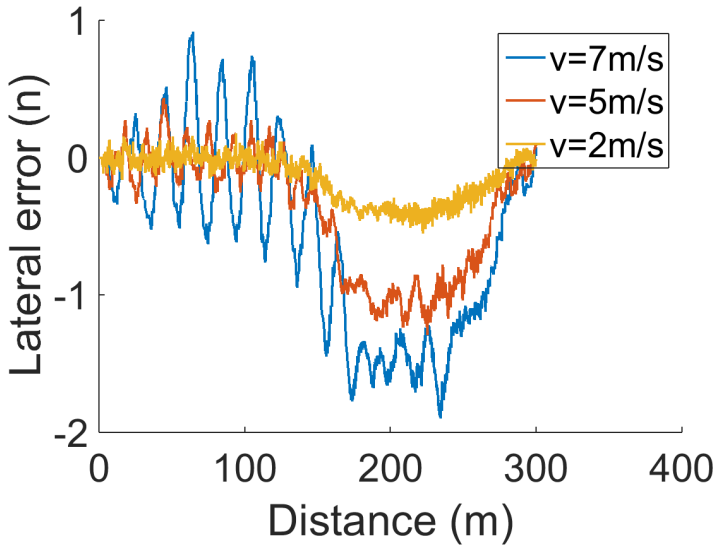

Figure 34: $Stanley_{V1}$ in simulation environment, tuned with orientation noise and without feed forward

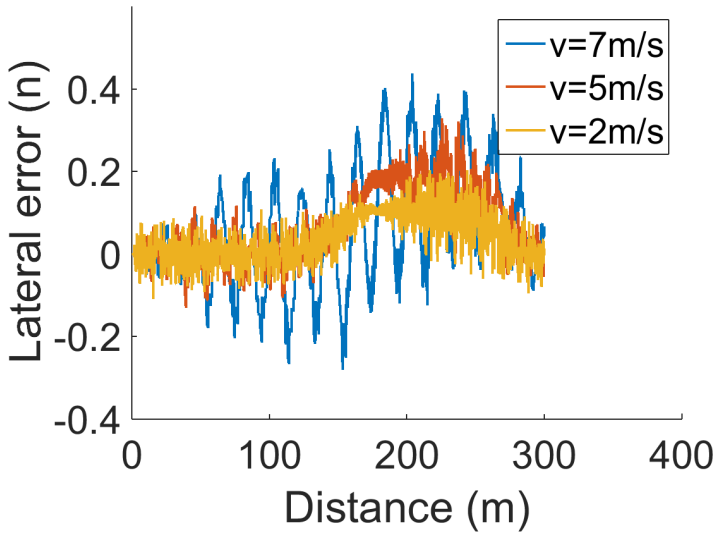

Figure 35: $Stanley_{V2}$ in simulation environment tuned with feed forward but without orientation noise

A.3.2 Experimental testing

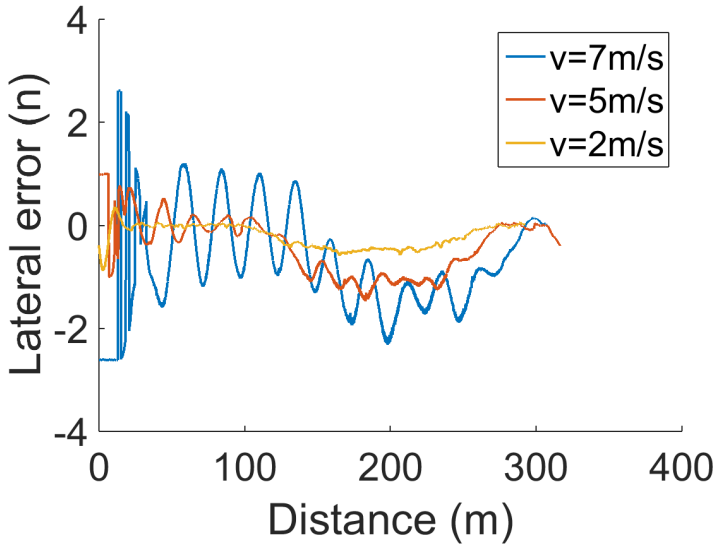

Figure 36: $Stanley_{V1}$ implemented in HX without feed forward

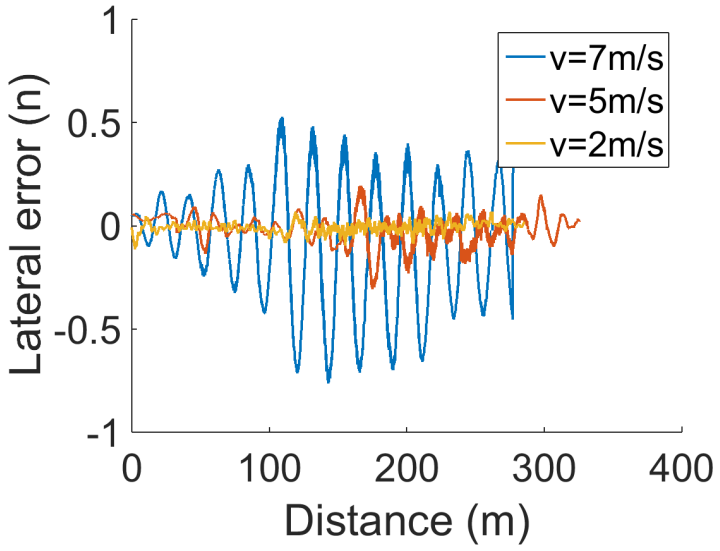

Figure 37: $Stanley_{V2}$ implemented in HX with feed forward

B Gain values

B.1 Fitness function

$$\begin{aligned} Fitness =& \omega_1 \sigma(d_{err}) \frac{1}{S_f} + \omega_2 \text{mean} \left| d_{err} \right| \frac{1}{C_f} \\ &+ \omega_3 \sum \left| \frac{\mathrm{d}}{\mathrm{d}t} d_{err} \right| \frac{1}{D_f} + \omega_4 \text{mean} \left| \theta_{err} \right| \frac{1}{O_{0f}} \\ &+ \omega_5 \sigma(\theta_{err}) \frac{1}{O_{1f}} + \omega_6 \max \left| \theta_{err} \right| \frac{1}{O_{2f}} + \max \left| d_{err} \right| \end{aligned}$$

$$\omega_1 = 1$$
$$\omega_2 = 1$$
$$\omega_3 = 1$$
$$\omega_4 = 1$$
$$\omega_5 = 1$$
$$\omega_6 = 1$$
$$\omega_7 = 5$$
$$S_f = 10$$
$$C_f = 10$$
$$D_f = 100$$
$$O_{0f} = 5$$
$$O_{1f} = 5$$
$$O_{2f} = 15$$

References

[1] A. Shaout and M. Jarrah, "Cruise control technology review," *Computers & electrical engineering*, vol. 23, no. 4, pp. 259–271, 1997.

[2] M. Turk, D. Morgenthaler, K. Gremban, and M. Marra, "Video road-following for the autonomous land vehicle," in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4. IEEE, 1987, pp. 273–280.

[3] E. D. Dickmanns and A. Zapp, "Autonomous high speed road vehicle guidance by computer vision," in *International Federation of Automatic Control. World Congress (10th). Automatic control: world congress.*, vol. 1, 1988.

[4] T. Luettel, M. Himmelsbach, and H.-J. Wuensche, "Autonomous ground vehiclesconcepts and a path to the future," *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1831–1839, 2012.

[5] C. Li, J. Wang, X. Wang, and Y. Zhang, "A model based path planning algorithm for self-driving cars in dynamic environment," in *Chinese Automation Congress (CAC), 2015*. IEEE, 2015, pp. 1123–1128.

[6] K. Jo, J. Kim, D. Kim, C. Jang, and M. Sunwoo, "Development of autonomous carpart i: Distributed system architecture and development process," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 12, pp. 7131–7140, 2014.

[7] M. Gerla, E.-K. Lee, G. Pau, and U. Lee, "Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds," in *Internet of Things (WF-IoT), 2014 IEEE World Forum on*. IEEE, 2014, pp. 241–246.

[8] H. Al Najada and I. Mahgoub, "Autonomous vehicles safe-optimal trajectory selection based on big data analysis and predefined user preferences," in *Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), IEEE Annual*. IEEE, 2016, pp. 1–6.

[9] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations," *Transportation Research Part A: Policy and Practice*, vol. 77, pp. 167–181, 2015.

[10] S. nc, J. Ploeg, N. van de Wouw, and H. Nijmeijer, "Cooperative adaptive cruise control: Network-aware analysis of string stability," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1527–1537, Aug 2014.

[11] V. Butakov and P. Ioannou, "Driving autopilot with personalization feature for improved safety and comfort," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Sept 2015, pp. 387–393.

[12] C. C. Penna and F. W. Geels, "Climate change and the slow reorientation of the american car industry (1979–2012): An application and extension of the dialectic issue lifecycle (dilc) model," *Research Policy*, vol. 44, no. 5, pp. 1029–1048, 2015.

[13] E. Helmers and P. Marx, "Electric cars: technical characteristics and environmental impacts," *Environmental Sciences Europe*, vol. 24, no. 1, p. 14, 2012.

[14] K. M. Tan, V. K. Ramachandaramurthy, and J. Y. Yong, "Integration of electric vehicles in smart grid: A review on vehicle to grid technologies and optimization techniques," *Renewable and Sustainable Energy Reviews*, vol. 53, pp. 720–732, 2016.

[15] A. Ghaderi, "Commercial vehicle electrification," *IAIC-2016*, 2016.

[16] M. Weiss, P. Dekker, A. Moro, H. Scholz, and M. K. Patel, "On the electrification of road transportation–a review of the environmental, economic, and social performance of electric two-wheelers," *Transportation Research Part D: Transport and Environment*, vol. 41, pp. 348–366, 2015.

[17] S. Bakker and J. Farla, "Electrification of the car–will the momentum last?: Introduction to the special issue," 2015.

[18] M. L. Sichitiu and M. Kihl, "Inter-vehicle communication systems: a survey," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 2, 2008.

[19] W. L. Jin and W. W. Recker, "An analytical model of multihop connectivity of inter-vehicle communication systems," *IEEE Transactions on Wireless Communications*, vol. 9, no. 1, pp. 106–112, January 2010.

[20] S. TSUGAWA, "{ISSUES} {AND} {RECENT} {TRENDS} {IN} {VEHICLE} {SAFETY} {COMMUNICATION} {SYSTEMS}," {*IATSS*} *Research*, vol. 29, no. 1, pp. 7 – 15, 2005. [Online]. Available: //www.sciencedirect.com/science/article/pii/S0386111214601138

[21] S.-H. Yu, O. Shih, H.-M. Tsai, N. Wisitpongphan, and R. D. Roberts, "Smart automotive lighting for vehicle safety," *IEEE Communications Magazine*, vol. 51, no. 12, pp. 50–59, 2013.

[22] G. Meyer, "Synergies of connectivity, automation and electrification of road vehicles," in *Road Vehicle Automation 3*. Springer, 2016, pp. 187–191.

[23] S. Dersten, P. Wallin, J. Fröberg, and J. Axelsson, "Analysis of the information needs of an autonomous hauler in a quarry site," in *System of Systems Engineering Conference (SoSE), 2016 11th*. IEEE, 2016, pp. 1–6.

[24] Caterpillar, "Caterpillar Year In Review 2015," http://www.caterpillar.com/en/investors/year-in-review.html, February 2016.

[25] Komatsu, "KOMATSU REPORT 2015," http://www.komatsu.com/CompanyInfo/ir/annual/annual2015.html, July 2016.

[26] Hitachi, "Annual Report 2015," http://www.hitachi.com/IR-e/library/integrated/2015/ar2015e.pdf, March 2016.

[27] Volvo Group, "The fourth quarter and full year 2015," http://www.volvogroup.com/en-en/events/2016/feb/interim-fourth-quarter-2015.html, February 2016.

[28] B. Creagh. (2017) Caterpillar to expand range of autonomous trucks. [Online]. Available: http://web.archive.org/web/20080207010024/http://www.808multimedia.com/winnt/kernel.htm

[29] Komatsu. Autonomous haulage system. [Online]. Available: http://www.komatsuamerica.com/innovation/autonomous-navigation

[30] Hitachi. (2015) Hitachi to develop autonomous haulage system. [Online]. Available: https://www.hitachicm.eu/press-center/hitachi-to-develop-autonomous-haulage-system/

[31] S. Bennett, *A history of control engineering, 1930-1955*. IET, 1993, no. 47.

[32] K.-E. Arzén, "A simple event-based pid controller," in *Proc. 14th IFAC World Congress*, vol. 18, 1999, pp. 423–428.

[33] L. Asplund, "Robotik," pp. 386–388, 2011, ISBN:978-91-47-01941-0.

[34] R. Marino, S. Scalzi, and M. Netto, "Nested pid steering control for lane keeping in autonomous vehicles," *Control Engineering Practice*, vol. 19, no. 12, pp. 1459–1467, 2011.

[35] A. Visioli, "Practical pid control," pp. 8–9, 2006, ISBN:978-1-84628-585-1.

[36] K. J. Åström, H. Panagopoulos, and T. Hägglund, "Design of pi controllers based on non-convex optimization," *Automatica*, vol. 34, no. 5, pp. 585–601, 1998.

[37] J. Lee, W. Cho, and T. F. Edgar, "Multiloop pi controller tuning for interacting multivariable processes," *Computers & chemical engineering*, vol. 22, no. 11, pp. 1711–1723, 1998.

[38] G. M. Hoffmann, C. J. Tomlin, M. Montemerlo, and S. Thrun, "Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing," in *2007 American Control Conference*, July 2007, pp. 2296–2301.

[39] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann *et al.*, "Stanley: The robot that won the darpa grand challenge," *Journal of field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[40] R. C. Rafaila and G. Livint, "H-infinity control of automatic vehicle steering," in *2016 International Conference and Exposition on Electrical and Power Engineering (EPE)*, Oct 2016, pp. 031–036.

[41] G. S. Glad, Torkel and L. Ljung, "Digital styrning kurskompendium."

[42] F. Yakub, A. Abu, S. Sarip, and Y. Mori, "Study of model predictive control for path-following autonomous ground vehicle control under crosswind effect," *Journal of Control Science and Engineering*, vol. 2016, 2016.

[43] W. Xi and J. S. Baras, "Mpc based motion control of car-like vehicle swarms," in *2007 Mediterranean Conference on Control Automation*, June 2007, pp. 1–6.

[44] P. Falcone, H. Eric Tseng, F. Borrelli, J. Asgari, and D. Hrovat, "Mpc-based yaw and lateral stabilisation via active front steering and braking," *Vehicle System Dynamics*, vol. 46, no. S1, pp. 611–628, 2008.

[45] R. Lenain, B. Thuilot, C. Cariou, and P. Martinet, "High accuracy path tracking for vehicles in presence of sliding: Application to farm vehicle automatic guidance for agricultural tasks," *Autonomous Robots*, vol. 21, no. 1, pp. 79–97, 2006. [Online]. Available: http://dx.doi.org/10.1007/s10514-006-7806-4

[46] J. M. Snider *et al.*, "Automatic steering methods for autonomous automobile path tracking," *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RITR-09-08*, 2009.

[47] A. Ollero, A. García-Cerezo, and J. Martinez, "Fuzzy supervisory path tracking of mobile reports," *Control Engineering Practice*, vol. 2, no. 2, pp. 313–319, 1994.

[48] A. Rodríguez-Castaño, G. Heredia, and A. Ollero, "Analysis of a gps-based fuzzy supervised path tracking system for large unmanned vehicles," in *Proceedings of the 4th IFAC International Symposium on Intelligent Components and Instruments for Control Applications (SICICA'00)*, 2000, pp. 141–146.

[49] C. M. Filho, D. F. Wolf, V. Grassi, and F. S. Osrio, "Longitudinal and lateral control for autonomous ground vehicles," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, June 2014, pp. 588–593.

[50] M. Negnevitsky and A. Wesley, "Artificial intelligence - a guide to intelligent systems," pp. 87–124, 2005, ISBN:0-321-20466-2.

[51] X. Wang, M. Fu, H. Ma, and Y. Yang, "Lateral control of autonomous vehicles based on fuzzy logic," *Control Engineering Practice*, vol. 34, pp. 1–17, 2015.

[52] A. Visioli, "Tuning of pid controllers with fuzzy logic," *IEE Proceedings - Control Theory and Applications*, vol. 148, no. 1, pp. 1–8, Jan 2001.

[53] M. Green and D. Limebeer, *Linear robust control*, 1995, ISBN:9780131022782.

[54] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*, 2005, ISBN:9780470011683.

[55] H. K. Khalil, *Nonlinear Systems, Third Edition*, 2002, ISBN:0-13-067389-7.

[56] C. Cariou, R. Lenain, B. Thuilot, and P. Martinet, "Adaptive control of four-wheel-steering off-road mobile robots: Application to path tracking and heading control in presence of sliding," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2008, pp. 1759–1764.

[57] P. G. Trepagnier, J. Nagel, P. M. Kinney, C. Koutsougeras, and M. Dooner, "Kat-5: Robust systems for autonomous vehicle navigation in challenging and unknown terrain," *Journal of Field Robotics*, vol. 23, no. 8, pp. 509–526, 2006.

[58] S. M. E. Goodwin Graham C., Graebe Stefan F., "Control system design," pp. 167–169, 2000, ISBN:978-0139586538.

[59] K. D. Young, V. I. Utkin, and U. Ozguner, "A control engineer's guide to sliding mode control," in *Variable Structure Systems, 1996. VSS'96. Proceedings., 1996 IEEE International Workshop on*. IEEE, 1996, pp. 1–14.

[60] T. Hiraoka, O. Nishihara, and H. Kumamoto, "Automatic path-tracking controller of a four-wheel steering vehicle," *Vehicle System Dynamics*, vol. 47, no. 10, pp. 1205–1227, 2009. [Online]. Available: http://dx.doi.org/10.1080/00423110802545919

[61] S. Dominguez, A. Ali, G. Garcia, and P. Martinet, "Comparison of lateral controllers for autonomous vehicle: Experimental results," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2016, pp. 1418–1423.

[62] S. Chaib, M. S. Netto, and S. Mammar, "H infin;, adaptive, pid and fuzzy control: a comparison of controllers for vehicle lane keeping," in *IEEE Intelligent Vehicles Symposium, 2004*, June 2004, pp. 139–144.

[63] R. Marino and P. Tomei, *Nonlinear control design: geometric, adaptive and robust*. Prentice Hall International (UK) Ltd., 1996.

[64] P. Corke, "Robotics, vision and control," p. 27, 2011, ISBN:978-3-642-20143-1.

[65] H. Nyquist, "Regeneration theory," *Bell Labs Technical Journal*, vol. 11, no. 1, pp. 126–147, 1932.

[66] A. Micaelli and C. Samson, "Trajectory tracking for unicycle-type and two-steering-wheels mobile robots," Ph.D. dissertation, INRIA, 1993.

[67] E. Lantto *et al.*, *Robust control of magnetic bearings in subcritical machines*. Helsinki University of Technology, 1999.

[68] J. Zhang, J. Zhuang, H. Du *et al.*, "Self-organizing genetic algorithm based tuning of pid controllers," *Information Sciences*, vol. 179, no. 7, pp. 1007–1018, 2009.

[69] J. H. Holland, "Adaptation in natural and artificial systems. an introductory analysis with application to biology, control, and artificial intelligence," *Ann Arbor, MI: University of Michigan Press*, 1975.

[70] S. G. Kumar, R. Jain, N. Anantharaman, V. Dharmalingam, and K. Begum, "Genetic algorithm based pid controller tuning for a model bioreactor," *indian chemical engineer*, vol. 50, no. 3, pp. 214–226, 2008.

[71] J.-S. Kim, J.-H. Kim, J.-M. Park, S.-M. Park, W.-Y. Choe, and H. Heo, "Auto tuning pid controller based on improved genetic algorithm for reverse osmosis plant," *World Academy of Science, Engineering and Technology*, vol. 47, pp. 384–389, 2008.

[72] D. S. Pereira and J. O. Pinto, "Genetic algorithm based system identification and pid tuning for optimum adaptive control," in *Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME International Conference on*. IEEE, 2005, pp. 801–806.

[73] MatlabDocumenatation. (2017) Genetic algorithm. [Online]. Available: https://se.mathworks.com/help/gads/how-the-genetic-algorithm-works.html

[74] ——. (2017) Genetic algorithm options. [Online]. Available: https://se.mathworks.com/help/gads/ga.html