

Design and Implementation of a Path Following System for an Autonomous Vehicle

A.J. de Winter

Master of Science Thesis



Design and Implementation of a Path Following System for an Autonomous Vehicle

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Mechanical Engineering at Delft
University of Technology

A.J. de Winter

February 27, 2017

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of
Technology



The work in this thesis was conducted in collaboration with the Dutch Automated Vehicle Initiative. <http://davi.connekt.nl/>.



Copyright © Delft Center for Systems and Control (DCSC)
All rights reserved.



Abstract

At this moment a revolution is going on in the development of our vehicles. The driver is going to be replaced by the vehicle itself. To investigate this autonomy Delft University of Technology (TU Delft) is project leader of a partnership called Dutch Automated Vehicle Initiative (DAVI). This group of industry and academics are combining their efforts in development and implementation of autonomous driving systems. The aims of the initiative is to investigate, improve and demonstrate automated driving on public roads. One of the DAVI projects the TU Delft is currently working on is the development of an autonomous vehicle.

This autonomous vehicle is designed in different stages with different systems. Apart from observing the surroundings of the autonomous vehicle and planning a trajectory it is also required to have a system that is capable of controlling the movement of the vehicle. In this thesis such a system will be developed for the Toyota Prius of the DAVI project.

In contradiction to most Advanced Driving Assist Systems (ADAS) the developed system does not follow another vehicle, but instead it follows a trajectory that is represented as a list of path points. As part of this design a suitable vehicle model is selected with use of literature. Because this system will be based on the Toyota Prius it was required to find the vehicle model parameters from the actual vehicle. This is done with use of measurements and optimisation.

The path following system was split into two control parts, lateral controller for steering and a longitudinal controller for acceleration. These controllers are designed with the aim of real implementation via LabVIEW from National Instruments. Therefore the real-time capabilities are discussed. The control algorithms are selected based on the expected requirements for computational power.

After the construction of the path follower system in LabVIEW the performance was tested with use of simulations. The tuning of the controller was investigated and the best performing control parameters were found. Then the system was implemented into the real vehicle and tests were conducted to see the difference between simulation and real world. The results of these tests were then evaluated. The distance error between the vehicle and the path was in the real test within 15.8cm . It was proven that the system is able to perform in real-time with a satisfying performance in terms of stability, comfort and trajectory tracking.

Table of Contents

| | |
|---|-----------|
| Acknowledgements | xi |
| 1 Introduction | 1 |
| 1-1 The Vehicle | 2 |
| 1-2 Requirements | 5 |
| 1-3 Objective | 5 |
| 1-4 Outline | 6 |
| 2 Vehicle Dynamics | 7 |
| 2-1 Coordinate System | 7 |
| 2-2 Lateral Dynamics | 8 |
| 2-2-1 The Design of The Model | 8 |
| 2-2-2 Stability | 11 |
| 2-3 Longitudinal Dynamics | 12 |
| 2-3-1 Tyre Forces and Angles | 12 |
| 2-3-2 The Design of The Model | 14 |
| 2-4 Steering Dynamics | 14 |
| 2-5 Discussion | 14 |
| 3 Vehicle Parameters | 15 |
| 3-1 Weight and Location Centre of Gravity | 15 |
| 3-2 Steering Wheel | 17 |
| 3-3 Optimisation | 17 |
| 3-3-1 Data Sets | 18 |
| 3-3-2 Fitting | 18 |
| 3-4 Discussion | 21 |

| | | |
|----------|---|-----------|
| 4 | Control | 23 |
| 4-1 | Controller Goals | 23 |
| 4-2 | Lateral Control | 24 |
| 4-2-1 | Comfort | 24 |
| 4-2-2 | Future Predictive Control | 25 |
| 4-2-3 | Tuning and Performance | 27 |
| 4-3 | Longitudinal Control | 27 |
| 4-3-1 | Comfort | 27 |
| 4-3-2 | Safety | 28 |
| 4-3-3 | PD Control | 29 |
| 4-4 | Discussion | 29 |
| 5 | System Design | 31 |
| 5-1 | LabVIEW | 31 |
| 5-2 | The Path | 32 |
| 5-2-1 | Generation of a Trajectory | 33 |
| 5-3 | The System | 34 |
| 5-4 | Perpendicular Distance to The Path | 35 |
| 5-4-1 | Curve Equation | 36 |
| 5-4-2 | Current en Future Error | 39 |
| 5-4-3 | Local to Global Position | 40 |
| 6 | Simulations | 41 |
| 6-1 | Testbed Paths | 41 |
| 6-2 | Tuning | 42 |
| 6-2-1 | Steering Parameter k_s | 43 |
| 6-2-2 | Look-Ahead Distance Parameter k_f | 46 |
| 6-2-3 | Steering Output | 49 |
| 6-3 | Variable Look-Ahead Distance L_f | 52 |
| 6-4 | Higher Velocity | 55 |
| 6-5 | Discussion | 58 |
| 7 | Real Tests | 59 |
| 7-1 | Frequency | 59 |
| 7-2 | Trajectory | 62 |
| 7-3 | Heading | 64 |
| 7-4 | Test Drive | 67 |
| 7-4-1 | Accuracy of GPS | 68 |
| 7-5 | Discussion | 71 |
| 7-5-1 | Path | 71 |
| 7-5-2 | Heading | 72 |
| 7-5-3 | Performance | 72 |

| | |
|---------------------------------|-----------|
| 8 Conclusion | 73 |
| 9 Further Work Proposals | 75 |
| A Proposed controller | 77 |
| Bibliography | 79 |
| Glossary | 81 |
| List of Acronyms | 81 |
| List of Symbols | 82 |

List of Figures

| | | |
|-----|---|----|
| 1-1 | Levels of automation defined by the Society of Automotive Engineers, or SAE | 2 |
| 1-2 | Schematic view of the autonomous vehicle system | 3 |
| 1-3 | The Toyota Prius that is used for this thesis | 4 |
| 2-1 | Global and local coordinate system | 7 |
| 2-2 | The dynamic bicycle model | 9 |
| 2-3 | ISO coordinate system of a tyre | 12 |
| 2-4 | Side view of a wheel under acceleration | 13 |
| 3-1 | The RDW | 16 |
| 3-2 | The lateral tyre force regions | 17 |
| 3-3 | The inputs of one dataset | 19 |
| 3-4 | The outputs of one dataset | 19 |
| 3-5 | Data fit with the the highest average vaf value | 20 |
| 4-1 | The future vehicle of the future prediction control algorithm | 26 |
| 4-2 | Velocity-dependent rang of accelerations | 28 |
| 5-1 | A visualisation of the path | 32 |
| 5-2 | Creating a continues line for the heading | 33 |
| 5-3 | the heading calculated from GPS and the heading calculated from the path | 34 |
| 5-4 | Schematic view of the path follower | 35 |
| 5-5 | The bicycle model and the perpendicular distance to the path | 36 |
| 5-6 | The heading error θ_c before and after conversion | 37 |
| 5-7 | The conversion from global to local coordinates | 38 |
| 5-8 | The same path section plotted in different ways | 39 |
| 5-9 | Schematic explanation of the path follower | 40 |

| | | |
|------|---|----|
| 6-1 | The path and simulation responses of variation k_s | 44 |
| 6-2 | Steering angle responses of variation k_s | 44 |
| 6-3 | Control input for different simulations of variation k_s | 45 |
| 6-4 | Evaluation data of different simulations of variation k_s | 45 |
| 6-5 | The path and simulation responses of variation k_f | 47 |
| 6-6 | Steering angle responses of variation k_f | 47 |
| 6-7 | Control input for different simulations of variation k_f | 48 |
| 6-8 | Evaluation data of different simulations of variation k_f | 48 |
| 6-9 | Steering angle responses for evaluation of steering output | 50 |
| 6-10 | Control input for different simulations for evaluation of steering output | 51 |
| 6-11 | Evaluation data of different simulations for evaluation of steering output | 51 |
| 6-12 | The path and simulation responses of variation L_f | 53 |
| 6-13 | Steering angle responses of variation L_f | 53 |
| 6-14 | Control input for different simulations of variation L_f | 54 |
| 6-15 | Evaluation data of different simulations of variation L_f | 54 |
| 6-16 | The path and simulation for higher velocity | 56 |
| 6-17 | Steering angle responses for higher velocity | 56 |
| 6-18 | Control input for different simulations for higher velocity | 57 |
| 6-19 | Evaluation data of different simulations for higher velocity | 57 |
| 7-1 | Steering angle responses for evaluation of different frequencies | 60 |
| 7-2 | Control input for evaluation of different frequencies | 61 |
| 7-3 | Evaluation data for different frequencies | 61 |
| 7-4 | Trajectory for the test drive | 63 |
| 7-5 | Velocity profile for the test drive | 63 |
| 7-6 | The path and test responses for different headings | 65 |
| 7-7 | Steering angle responses for different headings | 65 |
| 7-8 | Control input for different headings | 66 |
| 7-9 | Evaluation data of different headings | 66 |
| 7-10 | The path, simulation responses and the response from the test drive | 69 |
| 7-11 | Controller output for simulations and test drive | 69 |
| 7-12 | Lateral control inputs for simulations and test drive | 70 |
| 7-13 | Evaluation data of simulations and test drive | 70 |
| 7-14 | Longitudinal control output and resulting velocity for simulations and test drive | 71 |
| A-1 | The preview structure of the proposed controller algorithm | 78 |
| A-2 | The different control modus and their calculations | 78 |

List of Tables

| | | |
|-----|---|----|
| 3-1 | Required vehicle parameters | 15 |
| 3-2 | Starting values for optimization | 20 |
| 3-3 | Acquired vehicle parameters | 21 |
| 4-1 | Threshold values of comfort | 24 |
| 4-2 | Acceleration analysis based on velocity region | 27 |
| 5-1 | Data of the path represented in a table | 32 |
| 6-1 | Minimum horizontal curve radius for constant velocity | 42 |
| 6-2 | Evaluation parameters for variation k_s | 43 |
| 6-3 | Evaluation parameters for variation k_f | 46 |
| 6-4 | Evaluation parameters for evaluation of steering output | 50 |
| 6-5 | Evaluation parameters for variation L_f | 52 |
| 6-6 | Evaluation parameters for higher velocity | 55 |
| 7-1 | Evaluation parameters for evaluation of different frequencies | 60 |
| 7-2 | Evaluation parameters for difference between simulation and real test | 68 |

Acknowledgements

After writing my thesis I can look back on a successful research with good results. My passion for automotive has always been there and with help of Dr.ir. Riender Happee I was able to do my MSc thesis research on this topic.

Special thanks to my supervisor Dr.ir. Simone Baldi who kept me in the right track with the biweekly meetings and provided me with the necessary support for the success of this research.

Further I would like to thank Ing. Tom Dalhuisen who, together with me, kept the Toyota Prius on in line and helped during the measurements and the test drive. Without him it would not have been possible to have these results.

During my research I worked together with Daniel den Hartog to organise and perform the tests to acquire the vehicle parameters. I would like to thank him for his collaboration and wish him good luck with his MSc thesis.

Delft, University of Technology
February 27, 2017

Alexander de Winter

“We did not come to fear the future. We came here to shape it.”

— *Barack Obama*

Chapter 1

Introduction

The last decade a lot of research is done into automation of vehicles. What started as small driving assist systems, like cruise control or a lane departure warning system, expanded rapidly to systems that are capable of stopping the vehicle in critical situations. Other Advanced Driving Assist Systems (ADAS), are able to keep the vehicle in its own lane and regulate the vehicles velocity from 0 to 130 kilometre per hour with adaptive cruise control.

Even though the vehicles are loaded with sensors, computers and actuators the drivers still do most of the driving themselves. Why are we able to have planes flying around completely automatic but are we not yet able to have self driving cars? One reason for this is that there are not as many planes in the sky as cars on the road. There is much more interaction with other objects on the road than there is in the sky. The complexity of the self driving vehicle is thus much more complicated than a autonomous plane.

Nevertheless the technology is becoming more and more advanced. In figure 1-1 a scheme of automation is shown. This scheme divided the level of automation into 5 levels and one level of no automation. The Society of Automotive Engineers, or SAE, has defined these levels of automation.

Current ADAS systems that are on the market are in level 1 of automation, and some rare systems in level 2. The level 2 automation ADAS systems are adaptive cruise control with lane keeping, but in these systems the human driver is still monitoring the driving environment and should be ready to intervene if the system fails. More advanced levels of automation systems are still under development.

Why is this research conducted, what is the huge benefit of autonomous vehicles? Every year merely in Europe, more than 40.000 casualties and 1.4 million injuries are caused by vehicle-related accidents [1]. The passive safety of the passenger cars is highly developed and therefore has almost reached its limit. In the active safety part there is however room for improvement. These active safety ADAS systems have the potential to significantly reduce the number of road accidents, but the best solution to overcome the huge number of injuries and casualties is to bypass the human driver. Most of the accidents with vehicles occur because of human error, if the human task of driving could be replaced by a machine than these accidents could be prevented.

| SAE Level | Name | Narrative Definition | Execution of Steering/ Acceleration/ Deceleration | Monitoring of Driving Environment | Fallback Performance of Dynamic Driving Task | System Capability (Driving Modes) |
|---|------------------------|--|---|-----------------------------------|--|-----------------------------------|
| <i>Human driver monitors the driving environment</i> | | | | | | |
| 0 | No Automation | the full-time performance by the <i>human driver</i> of all aspects of the <i>dynamic driving task</i> , even when enhanced by warning or intervention systems | Human driver | Human driver | Human driver | n/a |
| 1 | Driver Assistance | the <i>driving mode</i> -specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the <i>human driver</i> perform all remaining aspects of the <i>dynamic driving task</i> | Human driver and system | Human driver | Human driver | Some driving modes |
| 2 | Partial Automation | the <i>driving mode</i> -specific execution by one or more driver assistance systems of both steering and acceleration/deceleration using information about the driving environment and with the expectation that the <i>human driver</i> perform all remaining aspects of the <i>dynamic driving task</i> | System | Human driver | Human driver | Some driving modes |
| <i>Automated driving system ("system") monitors the driving environment</i> | | | | | | |
| 3 | Conditional Automation | the <i>driving mode</i> -specific performance by an <i>automated driving system</i> of all aspects of the <i>dynamic driving task</i> with the expectation that the <i>human driver</i> will respond appropriately to a <i>request to intervene</i> | System | System | Human driver | Some driving modes |
| 4 | High Automation | the <i>driving mode</i> -specific performance by an <i>automated driving system</i> of all aspects of the <i>dynamic driving task</i> , even if a <i>human driver</i> does not respond appropriately to a <i>request to intervene</i> | System | System | System | Some driving modes |
| 5 | Full Automation | the full-time performance by an <i>automated driving system</i> of all aspects of the <i>dynamic driving task</i> under all roadway and environmental conditions that can be managed by a <i>human driver</i> | System | System | System | All driving modes |

© Copyright 2014 SAE

Figure 1-1: Levels of automation defined by the Society of Automotive Engineers, or SAE

1-1 The Vehicle

One autonomous system that is under development is the self driving vehicle of the Dutch Automated Vehicle Initiative (DAVI). The research of this thesis will be part of this development. DAVI is an initiative that develops high automated vehicles for research and demonstrations on public roads. The research tries to proof the safety and focusses on human factors in automated driving. DAVI is a public private partnership between many different companies and institutes, the core group consists of the TU Delft, TNO, the RWD and Connekt.

One project of TU Delft and DAVI is a self driving vehicle for the public road. The vehicle is a Toyota Prius that is equipped with sensors and radar to view its surroundings. With these sensors and radars it is possible for the vehicle to have 360 degrees of sensing, this means everything that happens around the vehicle can be detected. Inside the vehicle there is also a system that is capable of controlling the steering, acceleration and brakes of the vehicle. Next to that an advanced Global Positioning System (GPS) is installed.

The autonomous vehicle of DAVI can be divided in a number of different systems. This is schematically shown in Figure 1-2. Part of the autonomous vehicle are:

- The tracker
- The eHorizon system
- The path planner
- The Move-Box
- The path follower

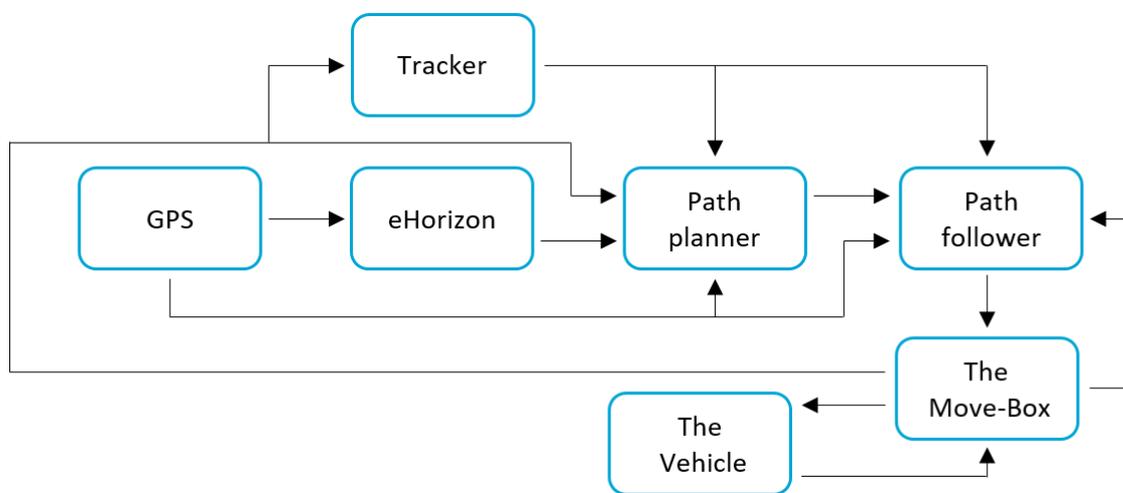


Figure 1-2: Schematic view of the autonomous vehicle system

The tracker is the system that uses the radars and cameras that are mounted on the vehicle. This system detects the other road users and objects around the vehicle.

The eHorizon system is capable of placing the vehicle inside a HD map with use of GPS then it can verify this location with use of camera images.

The path planner can be described as a highly advanced navigation system that doesn't only plan the route from A to B but also plans the trajectory. This trajectory will be a very detailed path which gives the desired position and velocity for the autonomous vehicle. This system needs to be able to change the trajectory when the surrounding changes, for example when other vehicles merge in front of the autonomous vehicle, or when pedestrians cross the road.

The Move-Box is the hardware-system that is located inside the vehicle. This system is the connection between a PCI eXtension for Instrumentation (PXI) and the vehicles board computer. This box ensures that there can be data transferred from vehicle to the PXI. The data can be transferred with a frequency of maximal 25 Hertz, this is the limit on the refresh rate of the sensors. Next to transferring information, this system is also capable of giving inputs to the vehicle. This can happen because the Move-Box is connected to the internal Adaptive Cruise Control (ACC) of the vehicle and therefore it is possible to control the velocity.

Next to the velocity it is also possible of controlling the steering-wheel angle. The development of this Move-Box system is done at TNO, they installed this complete system into the Toyota Prius.

The inputs of the Move-Box that will be used in this thesis are:

- The desired steering angle.
- The desired longitudinal acceleration.

The exact working of the Move-Box is unknown because it is not designed or installed at the TU delft itself and there is no full documentation available. Therefore it will be considered as a black box with known in- and outputs.



Figure 1-3: The Toyota Prius that is used for this thesis

The GPS that is installed in this vehicle is an advanced system from a company named Advanced Navigation. The GPS system is called Spatial Dual and consists out two antenna that are mounted on the roof of the vehicle as is shown in Figure 1-3. This system is much more precise than regular GPS systems.

The final system is the path follower. This system has as input the desired trajectory from the path planner and has the ability of making the vehicle actually follow this trajectory by controlling the acceleration and steering capabilities. Most ADAS systems follow other vehicles. The path follower is different, because instead of following other vehicles it follows a path that is represented as a list of values. This path follower will be the focus of this thesis.

1-2 Requirements

For the design of the path follower the following requirements apply:

- Compatible with soft- and hardware of the autonomous vehicle.
- Includes a control system for lateral and longitudinal control.
- Capable of running real-time.

The last requirement means that every calculation can be done within a certain time limit. For example determining the steering wheel position within a fraction of a second, not once but all the time. If the system is not able to calculate the desired steering wheel position in time then the controller will be unsafe. For example you do not want a autonomous vehicle that is driving on the highway and suddenly can not calculate the steering angle for the upcoming turn.

1-3 Objective

With the requirements for the path follower in mind the main aim of this MSc thesis can be defined as follows:

Designing and implementing a path following system for an autonomous vehicle that will follow a predefined trajectory in real-time.

This main objective can be divided into different sub objectives that are necessary to reach the main objective:

- Determining the necessary vehicle dynamics.
- Determining the parameters necessary for this vehicle model from the real car.
- Design of a controller for path following.
- Design of path following system.
- Tuning and evaluation of the controller in simulation.
- Tuning and evaluation of the controller in a real test.

In this thesis the other systems that are part of the DAVI system will not be used, because they are still in development. In this thesis a trajectory will be constructed to simulate and test the path follower. This also means that research on collision avoidance will not be part of this thesis because the tracker is not fully functional yet.

In this research the maximum velocity of the vehicle that will be considered is 50km/h . Although the system would be capable of higher velocities they are not the area of this research. However at the end of this thesis a lateral controller is proposed that should increase performance at higher velocities

1-4 Outline

In the beginning of this thesis the different aspects of vehicle dynamics that are useful for this thesis are introduced. The models that will be used throughout this thesis are explained and evaluated.

In chapter 3 it will be described how the different parameters that are required for the models were acquired. This includes a description of tests with the vehicle that were performed.

Chapter 4 will introduce the different control aspects of this thesis. First the lateral controller is introduced and then the longitudinal controller will be explained.

After the introduction of the control algorithms the design of the path follow system is explained in chapter 5. This chapter will introduce the program that is used in this thesis, it will describe the design of the path and the rest of the system that will be used to get the required inputs for the control algorithms. Chapter 6 will contain simulations of the path follower and chapter 7 contains a real test on the autonomous vehicle.

The final two chapters of this thesis will be a conclusion of the work and a description of the future work proposals.

In appendix A the previously described higher velocity lateral controller will be shown.

Vehicle Dynamics

In this chapter the vehicle model used in this thesis will be discussed. It will be explained why this model is chosen for the design of a path following controller and what the benefits and drawbacks of this model are. The coordinate systems that is used throughout the thesis will be introduced, and the stability of the model will be evaluated. At the end of this chapter the steering dynamics of the steering wheel are introduced.

2-1 Coordinate System

There are two coordinate systems that are used in this thesis, one is the global earth-fixed coordinate system (X, Y, Z) and the other is a local coordinate system (X_l, Y_l, Z_l) . The global system is fixed to the earth at any convenient location and is used to define the position and the orientation of the vehicle. The origin of the local system is fixed to the vehicle's centre of gravity and moves as the vehicle moves. This body-fixed system is used to define the motion of the vehicle. In Figure 2-1 these coordinate systems are shown.

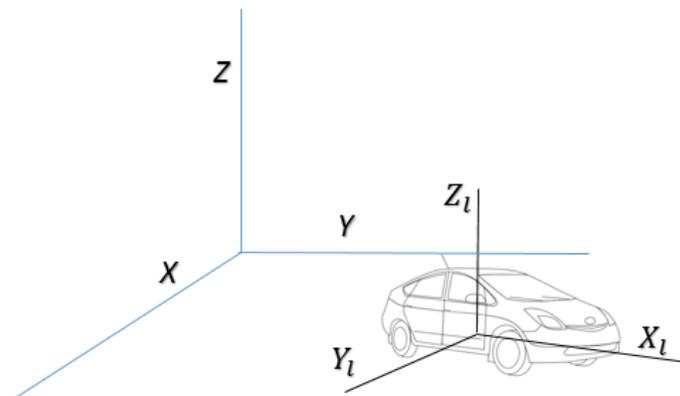


Figure 2-1: Global and local coordinate system

2-2 Lateral Dynamics

The model that is used for lateral dynamics is the bicycle model. This model is chosen because it is the most commonly used for path following. As the name suggest it is a model that looks like a bicycle. The model combines the left and right wheels into a pair of single wheels at the centre of the front and rear axle. The model is based upon the following assumptions [4] [5]:

- The vehicles longitudinal velocity is taken as a time-variant parameter. This means that the acceleration in longitudinal direction is set to zero, thereby neglecting tire-slip in longitudinal direction.
- The vehicle is moving on a flat surface.
- The tyres response in the linear range.
- The steering input to the front tyres are small and the thereby corresponding slip angles are also small.
- The vehicle structure, including the suspension system, is rigid.
- The track width is small compared to the radius of the turn, thereby neglecting the small angle difference between the front wheels.
- The aerodynamic forces are negligible.

As is shown there are a lot of assumptions made in this model. One might think that all these assumptions will result in a model that does not represent the real world and therefore the controllers made with this model will not work properly. Vehicle dynamics are very complicated and high fidelity models are very non-linear, these non-linear models will represent the real-world in more situations. But the question is if that is necessary? Most of the non-linearities act in extreme cases as high-speed steering manoeuvres at for example racing-tracks or extreme braking. These situations will not be encountered in the normal driving situation.

It is indeed true that the bicycle model will not represent the real world in every situation but under normal driving situations it will represent the system well enough for the purpose of this thesis.

2-2-1 The Design of The Model

The bicycle model is created with use of Newton's second law of motion, the equations for lateral and yaw motions can be written as:

$$\begin{aligned} ma_y &= \sum F_y \\ I_z \dot{r} &= \sum M_z \end{aligned} \quad (2-1)$$

With

$$\begin{aligned} m &= \text{Vehicle mass} && [kg] \\ a_y &= \text{Lateral vehicle acceleration} && [m/s^2] \end{aligned}$$

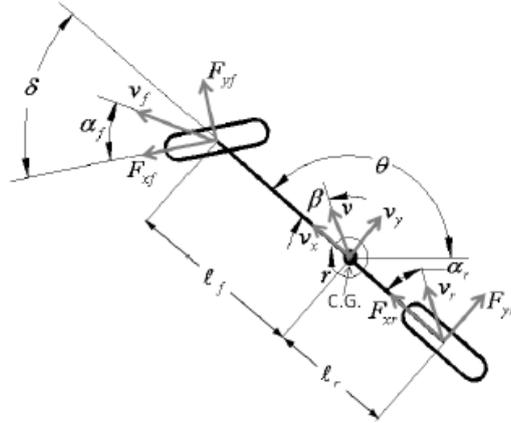


Figure 2-2: The dynamic bicycle model

From Figure 2-2 the sum of forces in the y-direction and the total moment about the z-axis can be observed. These sums are given as:

$$\begin{aligned}\sum F_y &= F_{yf} \cos(\delta) + F_{yr} \\ \sum M_z &= l_f F_{yf} \cos(\delta) - l_r F_{yr}\end{aligned}$$

The acceleration in the y-direction a_y exists out of two parts, one is the acceleration \dot{v}_y and the other one is the influence of the rotation of the vehicle $v_x r$ with r given as the angular velocity about the z-axis also known as the yaw rate. This gives:

$$a_y = \dot{v}_y + v_x r \quad (2-2)$$

Substituting these values into Eq. (2-1) gives the equations of motion as:

$$\begin{aligned}m(\dot{v}_y + v_x r) &= F_{yf} \cos(\delta) + F_{yr} \\ I_z \dot{r} &= l_f F_{yf} \cos(\delta) - l_r F_{yr}\end{aligned} \quad (2-3)$$

The force generated by the front and rear wheels can be modelled linearly proportional to the slip angle, this results in the lateral force defined as shown in Eq. (2-4).

$$\begin{aligned}F_{yf} &= C_{\alpha f} \alpha_f \\ F_{yr} &= C_{\alpha r} \alpha_r\end{aligned} \quad (2-4)$$

With:

$$\begin{aligned}\alpha_f &= \text{Slip angle of the front tyre} && [rad] \\ \alpha_r &= \text{Slip angle of the rear tyre} && [rad] \\ C_{\alpha f} &= \text{Cornering stiffness of 2 front tyres} && [N/rad] \\ C_{\alpha r} &= \text{Cornering stiffness of 2 rear tyres} && [N/rad]\end{aligned}$$

The front and rear tyre slip is calculated with Eq. (2-5).

$$\begin{aligned}\alpha_f &= \delta - \tan^{-1} \left(\frac{v_y + l_f r}{v_x} \right) \\ \alpha_r &= -\tan^{-1} \left(\frac{v_y + l_f r}{v_x} \right)\end{aligned}\quad (2-5)$$

Substituting Eq. (2-3) and Eq. (2-4) into Eq. (2-5) results in:

$$\begin{aligned}m(\dot{v}_y + v_x r) &= C_{\alpha f} \left[\delta - \tan^{-1} \left(\frac{v_y + l_f r}{v_x} \right) \right] \cos(\delta) - C_{\alpha r} \tan^{-1} \left(-\frac{v_y + l_f r}{v_x} \right) \\ I_z \dot{r} &= l_f C_{\alpha f} \left[\delta - \tan^{-1} \left(\frac{v_y + l_f r}{v_x} \right) \right] \cos(\delta) + l_r C_{\alpha r} \tan^{-1} \left(-\frac{v_y + l_f r}{v_x} \right)\end{aligned}\quad (2-6)$$

The next step towards a model is to linearise the given equations of motion. This is done using the small angle assumption to Eq. (2-6). This will result in the following:

$$\begin{aligned}m(\dot{v}_y + v_x r) &= C_{\alpha f} \left[\delta - \left(\frac{v_y + l_f r}{v_x} \right) \right] + C_{\alpha r} \left(\frac{v_y + l_f r}{v_x} \right) \\ I_z \dot{r} &= l_f C_{\alpha f} \left[\delta - \left(\frac{v_y + l_f r}{v_x} \right) \right] - l_r C_{\alpha r} \left(\frac{v_y + l_f r}{v_x} \right)\end{aligned}$$

Solving these equations for \dot{v}_y and \dot{r} gives:

$$\begin{aligned}\dot{v}_y &= -\frac{C_{\alpha f} + C_{\alpha r}}{m v_x} v_y + \left[-v_x + \frac{l_r C_{\alpha r} - l_f C_{\alpha f}}{m v_x} \right] r + \frac{C_{\alpha f}}{m} \delta \\ \dot{r} &= \frac{l_r C_{\alpha r} - l_f C_{\alpha f}}{I_z v_x} v_y + \frac{-l_r^2 C_{\alpha r} - l_f^2 C_{\alpha f}}{I_z v_x} r + \frac{l_f C_{\alpha f}}{I_z} \delta\end{aligned}\quad (2-7)$$

Finally the linear dynamic bicycle model can be written in state space form as:

$$\begin{bmatrix} \dot{v}_y \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -\frac{C_{\alpha f} + C_{\alpha r}}{m v_x} & -v_x + \frac{l_r C_{\alpha r} - l_f C_{\alpha f}}{m v_x} \\ \frac{l_r C_{\alpha r} - l_f C_{\alpha f}}{I_z v_x} & \frac{-l_r^2 C_{\alpha r} - l_f^2 C_{\alpha f}}{I_z v_x} \end{bmatrix} \begin{bmatrix} v_y \\ r \end{bmatrix} + \begin{bmatrix} \frac{C_{\alpha f}}{m} \\ \frac{l_f C_{\alpha f}}{I_z} \end{bmatrix} \delta \quad (2-8)$$

2-2-2 Stability

In this subsection the stability of the Linear dynamic bicycle model is discussed. Because this is a linear system it is stable if the eigenvalues have no positive real parts. This means that all terms of Eq. (2-9) need to be positive.

$$\det(\lambda I - A) = \lambda^2 - (a_{11} + a_{22})\lambda + (a_{11}a_{22} - a_{21}a_{12}) = 0 \quad (2-9)$$

When Eq. (2-9) is filled with the terms it results in:

$$\det(\lambda I - A) = \lambda^2 \left(\frac{C_{\alpha f} + C_{\alpha r}}{mv_x} + \frac{l_r^2 C_{\alpha r} l_f^2 C_{\alpha f}}{I_z v_x} \right) \lambda + \left(\frac{C_{\alpha f} C_{\alpha r} (l_f^2 + l_r^2)}{I_z m v_x^2} + \frac{l_r C_{\alpha r} - l_f C_{\alpha f}}{I_z} \right) = 0$$

It is clear that the second term is always positive, and the only minus sign is in the third term. This means that all parts are positive if:

$$l_r C_{\alpha r} > l_f C_{\alpha f} \quad (2-10)$$

If $l_r C_{\alpha r} < l_f C_{\alpha f}$ the system can still be stable, but it depends on the forward velocity v_x . It is possible to find this critical forward velocity using the following equation [2]:

$$\frac{C_{\alpha f} C_{\alpha r} (l_f^2 + l_r^2)}{l_f C_{\alpha f} - l_r C_{\alpha r}} > m v_x^2$$

Using the fact that $L = l_f + l_r$ we get:

$$\frac{L^2 C_{\alpha f} C_{\alpha r}}{l_f C_{\alpha f} - l_r C_{\alpha r}} > m v_x^2 = \frac{L^2}{\frac{l_f}{C_{\alpha r}} - \frac{l_r}{C_{\alpha f}}} > m v_x^2$$

This results in the critical velocity of:

$$v_x < \sqrt{\frac{L^2}{m \left(\frac{l_f}{C_{\alpha r}} - \frac{l_r}{C_{\alpha f}} \right)}}$$

It is now possible to simplify this equation using the definition for the under-steer gradient shown in Eq. (2-11).

$$K_{us} = \frac{m}{L} \left(\frac{l_r}{C_{\alpha f}} - \frac{l_f}{C_{\alpha r}} \right) \quad (2-11)$$

This results in the final equation for the critical forward velocity shown in Eq. (2-12). The stability conditions are Eq. (2-10) and if that does not hold than Eq. (2-12).

$$v_x < \sqrt{\frac{L}{-K_{us}}} \quad (2-12)$$

2-3 Longitudinal Dynamics

As said in section 2-2 the longitudinal velocity is taken as a time-variant parameter in the bicycle model. This can be observed in Eq. (2-8) where the longitudinal velocity is just a parameter. Nevertheless it is possible to use another model for the longitudinal velocity, place this velocity in the bicycle model and have a total model for the vehicle.

A longitudinal vehicle model can also be called a tyre model because the most important component is the tyre. These tyres are complex dynamic components and have huge influence on behaviour of the vehicle. In the last decades an impressive amount of research is done regarding these tyres and their behaviour. A result of this research is several different types of tyre models with different characteristics.

2-3-1 Tyre Forces and Angles

The tyre coordinate system that will be used is the ISO axis system describes the forces and moments that are generated by a tyre. This system is shown in Figure 2-3. The tyre is

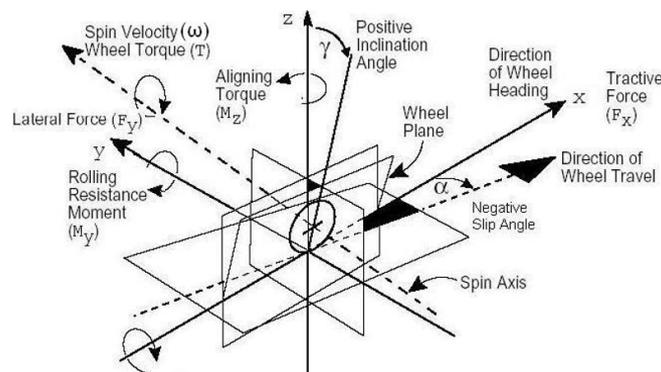


Figure 2-3: ISO coordinate system of a tyre

made of a viscoelastic rubber that also assists in isolating the vehicle from road disturbances. Therefore when the tyre comes in contact with the ground, a part of the structure deforms. The part of this structure that is in contact with the ground is called the contact patch. The longitudinal and lateral forces are mainly depended on the tyre elastic properties at small slip values, but if the slip increases than the contribution decreases and that of the friction between the tyre and the road increases.

In the previous section 2-2-1 the tyre side-slip angle was introduced. This angle is the difference between wheels velocity and the direction of the wheel which is shown in Figure 2-3. Next to side-slip there is also longitudinal wheel slip. This parameter is hugely important for any tyre model.

The longitudinal wheel slip λ is the difference in velocity between the vehicles body and the wheel velocity. This is better described with the side view of a wheel which is shown in Figure 2-4. The longitudinal velocity of the vehicles body is defined as v_x , this is at the centre of the wheel. If there is no acceleration or braking torque applied to the wheel it is free rolling at a velocity v_r . That means that the velocity of the wheel is equal to the velocity

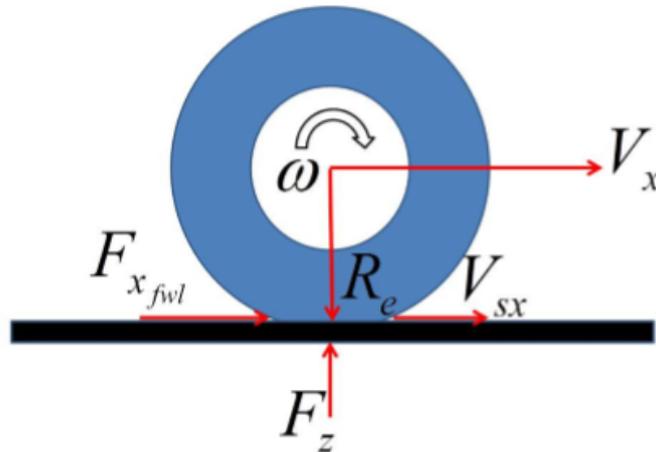


Figure 2-4: Side view of a wheel under acceleration

of the body as:

$$v_x = v_r$$

The velocity of the wheel v_r can be described as a function of the wheel rotations ω and the radius R_e as:

$$v_r = \omega R_e$$

When there is a braking torque the wheels start to decelerate, but because the tyres are viscoelastic the deceleration is not directly moved to the vehicles body. As a result the vehicles velocity v_x and the tyres velocity v_r are no longer the same. The difference between these velocities is defined as the longitudinal slip velocity shown in Figure 2-4 as v_{sx} .

$$v_{sx} = v_x - v_r$$

With this knowledge it is now possible to define the longitudinal slip λ as:

$$\begin{aligned} \lambda_b &= \frac{v_{sx}}{v_x} \\ \lambda_b &= \frac{v_x - v_r}{v_x} \\ \lambda_b &= \frac{v_x - \omega R_e}{v_x} \end{aligned} \quad (2-13)$$

λ_b is valid if the vehicle's velocity is higher than the rotational velocity of the wheel, thus under braking.

If the vehicle is accelerating it is the other way around, then the rotational velocity of the wheel is larger than the vehicles body. As a result the slip value during acceleration is defined as:

$$\lambda_a = \frac{\omega R_e - v_x}{\omega R_e} \quad (2-14)$$

λ_a and λ_b have a value between 0 and 1. If λ is 0 that means that the velocity of the wheel is exactly the same as the velocity of the body, if it is 1 then the wheel is fully locked, under braking, or it is fully spinning, under acceleration.

2-3-2 The Design of The Model

Two tyre models that are widely used are the so called semi-empirical magic formula [7] and the empirical Dugoff model [8]. Both of the models have their own limitations [4], the magic formula tyre model simulates the tyre forces in an excellent manner but it involves a number of unknown parameters that require extensive testing to be determined, and it is also computational heavier than the Dugoff tyre model. On the other hand the Dugoff model has shown errors at higher slip values when compared with experimental results.

As discussed in chapter 1 the output of the longitudinal controller should be the acceleration of the body. From the information of the previous section we can conclude that it is not possible to use one of the mentioned tyre models for this thesis. Those models can be used to describe the difference between body velocity and wheel velocity, thus the slip. The controllable parameter in these models is the wheel velocity. The problem is that with the set-up of this thesis it is not possible to give an acceleration value to the wheels. This is done by the Adaptive Cruise Control (ACC) controller of the Move-Box it self.

For design of the longitudinal controller a very simplistic longitudinal model is used. In simulation the longitudinal acceleration value will be integrated to the longitudinal velocity, and will be delayed with use of a transfer function. Then this velocity will be used in the bicycle model. This way it is possible to simulate the vehicle with different and changing longitudinal velocities.

2-4 Steering Dynamics

The controller will have an output of steering angle, not from the wheels themselves but the angle of the steering wheel. The steering wheel will be moved by an actuator that is build inside the Toyota Prius and is part of the set-up. This steering actuator will have some delay between receiving the desired steering angle and actually reaching that steering angle. To model this the steering actuator is approximated by a first-order lag element that is given in Eq. (2-15) [1].

$$\dot{\delta}_s = -\frac{\delta_s}{\tau_\delta} + \frac{\delta_{s_{des}}}{\tau_\delta} \quad (2-15)$$

In this equation τ_δ is the dynamic time constant, $\delta_{s_{des}}$ is the desired steering angle of the steering wheel and δ_s is the actual steering wheel angle. This function can be used in form of a transfer function, to do so the Laplace transform of the function of Eq. (2-15) is taken as shown in Eq. (2-16)

$$\delta_s(s) = \frac{\delta_{s_{des}}}{1 + \tau_\delta s} \quad (2-16)$$

2-5 Discussion

In this chapter the vehicle dynamic theory was reviewed. A suitable model to design a lateral controller has been found and explained. Different types of tire models are discussed and after researching these models it is concluded that they can not be used in this research due to the lac of information about the Move-Box system. A solution for this is proposed and explained. The next chapter will discuss how the parameters for these models were acquired.

Vehicle Parameters

The bicycle model described in the previous chapter exists out of a number of parameters. This chapter will discuss how the parameters for this model are found and what tests have been performed.

The parameters necessary for the complete model are shown in Table 3-1. These are all the values of the bicycle model Eq. (2-8), the steering ratio and a time constant. Some of these parameters can be calculated and some of them will need an experiment before a valid value is acquired.

Table 3-1: Required vehicle parameters

| Parameter | Value | Symbol | Unit |
|--|-------|----------------|----------|
| Length from front axle to CoG | ? | l_f | m |
| Length from rear axle to CoG | ? | l_r | m |
| Vehicle mass | ? | m | kg |
| Cornering stiffness of the front wheels | ? | C_{α_f} | N/rad |
| Cornering stiffness of the rear wheels | ? | C_{α_r} | N/rad |
| Moment of inertia around Z axis | ? | I_z | kg/m^2 |
| Steering ratio | ? | k_δ | - |
| Maximum steering angle of the steering wheel | ? | δ_{max} | rad |
| Dynamic steering time constant | ? | τ_δ | s |

3-1 Weight and Location Centre of Gravity

To find the length from the Centre of Gravity (CoG) to the front and rear axles it is first necessary to determine the location of the CoG. In this thesis it is assumed that this CoG is positioned in the centre of the vehicle broad wise. The model assumes that the left and right wheels are fit together and therefore the distribution of the weight should be equally divided between those wheels.

To locate the position of the CoG lengthways it is necessary to know the weight that rests on the front axle and the weight that rests on the rear axle. The vehicle manufacturer, in this case Toyota, gives information about the empty weight and maximum loadable weight. But no information about the distribution between front and rear axle. Even if this information was given it could easily be wrong for the current set-up because a lot of equipment has been added to the vehicle which may have changed the position of the centre of gravity.

Therefore the weight and weight distribution needed to be measured. To do this an appointment was made at the nearest Rijkdienst voor het Wegverkeer (RDW) inspection station: inspection station Schiedam. The RDW is the Dutch authority of road traffic. At the RDW it was possible to use their weighbridge which is shown in Figure 3-1b. With this bridge it was possible to measure the weight on the front and rear axle, which combined also gave the total mass of the vehicle.

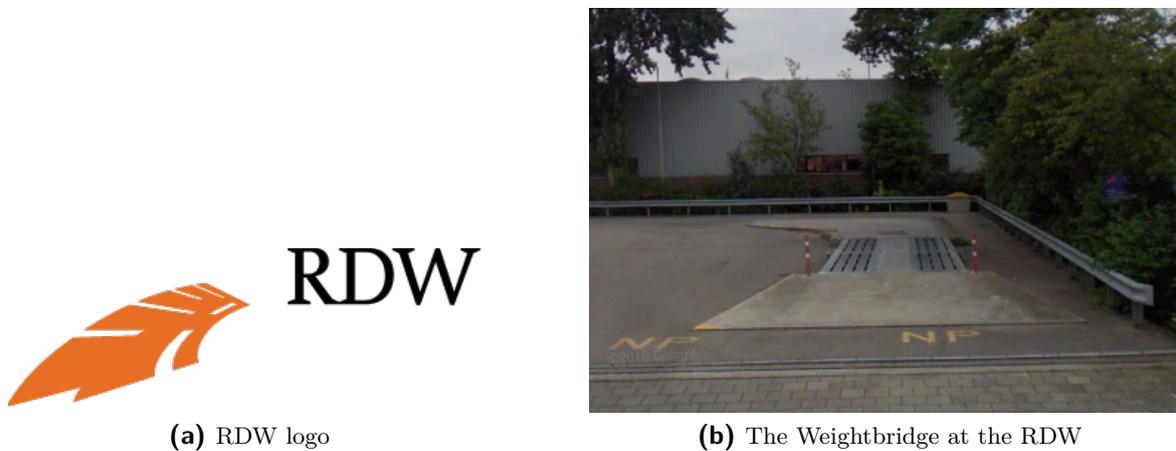


Figure 3-1: The RDW

After weighing the vehicle with 2 persons in the front chairs the front axle supported a weight of 950kg and the rear axle 640kg . The resolution of this weighbridge was 10 kg , which is 0.7% of the total weight.

These results will be used in the rest of the thesis. Now that the weight is known the lengths l_f and l_r can be calculated. From the documentation of the vehicle it is known that the wheelbase of the vehicle L is 2.7m . The calculations are done in Eq. (3-1).

$$l_f = \frac{L * m_r}{m} = \frac{2.7 * 640}{1590} = 1.0868\text{m} \quad (3-1)$$

Then the value for $l_r = L - l_f = 1.6132\text{m}$.

3-2 Steering Wheel

According to data-sheets of Toyota the steering ratio is 14.6:1 from steering wheel to wheel. With this information it is possible to calculate the maximum steering angle of the steering wheel when we know the maximum steering angle of the wheels.

To calculate the maximal steering angle the following equation is used:

$$\delta_{max_{wheels}} = \tanh \frac{L}{R - t/2} \quad (3-2)$$

Where R is the radius of the turning circle tested by the Dutch ANWB as $5.95m$. And t is the track width in meters.

$$\delta_{max_{wheels}} = \tanh \frac{2.7}{5,95 - 1.515/2} = 0.52rad$$

This multiplied by the steering ratio gives a maximal steering angle of the steering wheel as $7.592rad$.

The final steering wheel parameter is the dynamic steering time constant. Because it was not possible to perform a test in this stage of my research for this parameter a value from literature was used. The value used in simulation is $0.2s$ [1].

3-3 Optimisation

The parameters that are left to find are the cornering stiffness of the front and rear tyres and the moment of inertia around the Z axis. The cornering stiffness is a parameter that is only valid when applied within a linear range. This is best shown by Figure 3-2. In this picture C_y is the cornering stiffness C_α . F_y is the lateral tyre force and α is the slip angle.

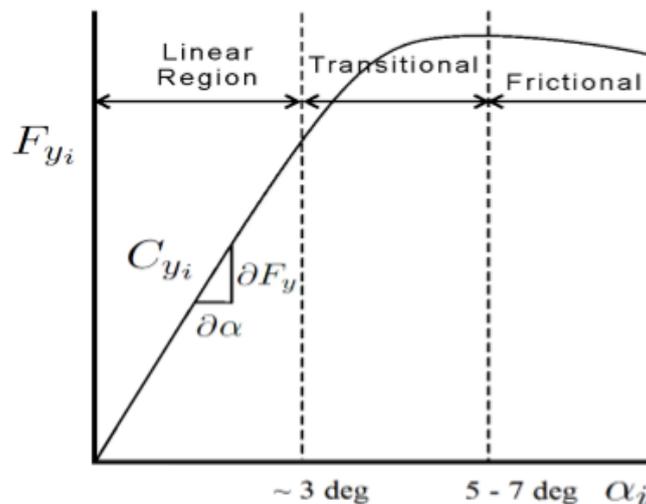


Figure 3-2: The lateral tyre force regions

The values of the cornering stiffness and the moment of inertia are found with use of optimisation. While driving the vehicle the data of the models inputs, the steering angle δ and the velocity v_x , are logged together with the model outputs, the lateral acceleration a_y and the yaw rate r . The model outputs are measured with the INS system of chapter 1. The model will be fit on the experiment data with a non-linear least squares method by using the algorithm `lsqnonlin` in Matlab, This is part of the `LTI System Identification Toolbox` [9]. This will result in the parameters that can be used in simulation.

3-3-1 Data Sets

To perform this test the vehicle was driven to Valkenburg Naval Air Base where enough space was available for the test. To find the parameters the following test was designed: a sinusoidal input would be put on the steering wheel with a constant velocity. Unfortunately the Move-Box couldn't be used for this test so it had to be done by hand. Therefore the velocity is not fully constant throughout the test and the input on the steering wheel isn't a perfect sinus. Four data sets were constructed: 2 with a constant velocity of 30km/h and 2 with a constant velocity of 40km/h the frequency of the sinus was 0.5Hz . The data sets consist out of 4 signals

- Input 1: The steering angle of the steering wheel.
- Input 2: The vehicle velocity.
- Output 1: The lateral acceleration.
- Output 2: the yaw rate.

In Figure 3-3 the inputs of one of these data sets is plotted. This is a set where 40km/h was driven but as is visible in the figure the velocity of the data set is lower. During the tests it was tried to keep the velocity of the vehicle at a constant value with use of the speedometer of the vehicle, but this speedometer had some deviation. The data that is shown in Figure 3-4 was logged with use of the Move-Box and this system is more accurate. Even though it is not exactly 40km/h or a constant velocity it doesn't influence the data fit that much because these fluctuations are used in the optimisation. In Figure 3-4 the measured outputs are plotted.

3-3-2 Fitting

When fitting a model to a dataset one will need a way to assess the fit of the parameters. This will be done with the the percentage variance accounted for (vaf) between two signals. When the two signals are the same the vaf value is 100% if they differ, the vaf value will be lower. This value can be calculated with Eq. (3-3). Where y is the real output and y_{est} is the estimated output.

$$vaf = \left(1 - \frac{\text{variance}(y - y_{est})}{\text{variance}(y)}\right) * 100 \quad (3-3)$$

All the datasets are optimised with use of the previously described algorithm `lsqnonlin`. This algorithm does require a starting point from which it tries to find an optimum. The starting values are shown in Table 3-2.

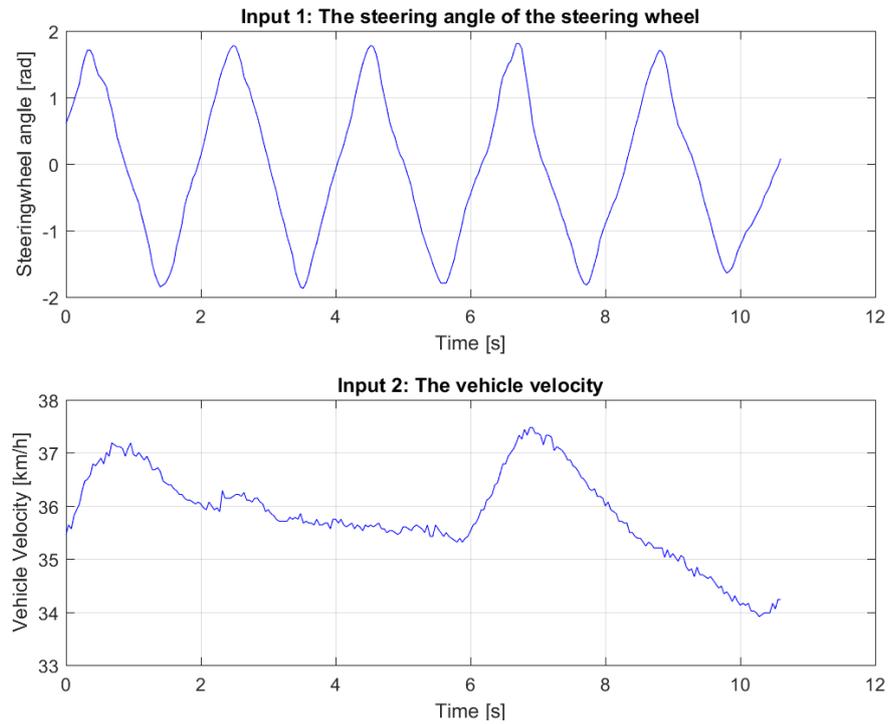


Figure 3-3: The inputs of one dataset

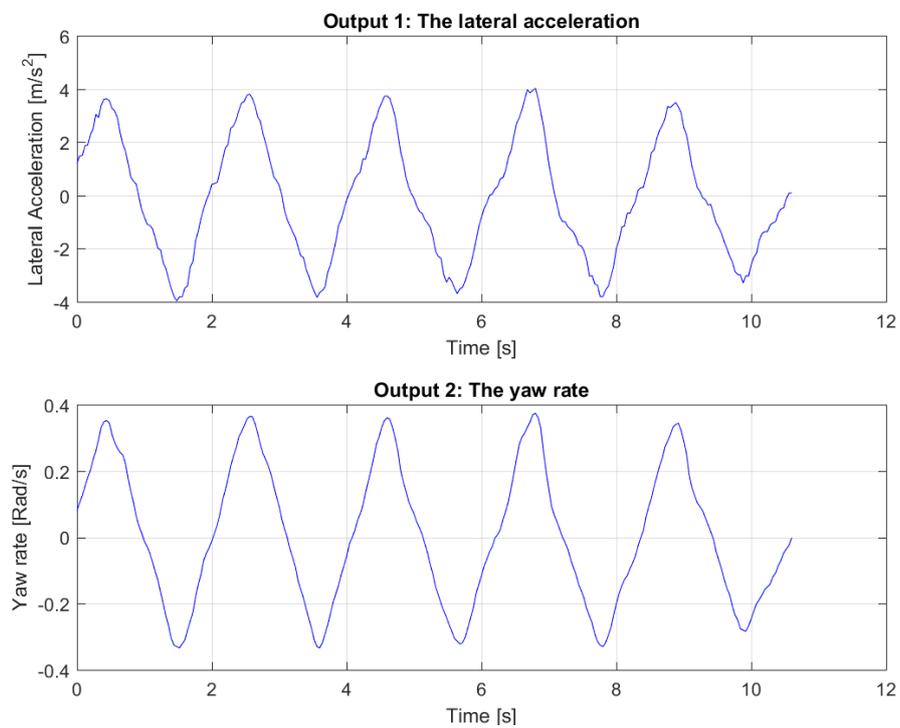


Figure 3-4: The outputs of one dataset

In the model there is a possibility for the rear tyres to have a different cornering stiffness than the front tyres. This can be the case when different tyres are fitted on the different axles. But with the Toyota Prius that is not the case, and therefore in the optimisation the cornering stiffness of the front and rear tyres are taken as the same value. That means that $C_{\alpha_f} = C_{\alpha_r} = C_{\alpha}$. Also one has to take into account that the bicycle model assumes that the left and rear tyres are combined. This means that the value of the cornering stiffness needs to be doubled. All the four data sets are fitted with use of `lsqnonlin`. the vaf value is

Table 3-2: Starting values for optimization

| Parameter | Value | Symbol | Unit |
|---|-------|----------------|----------|
| Cornering stiffness for one front wheel | 15000 | C_{α_f} | N/rad |
| Cornering stiffness for one rear wheel | 15000 | C_{α_r} | N/rad |
| Moment of inertia around Z axis | 700 | I_z | kg/m^2 |

calculated, and then the parameters are validated on the other datasets. For instance dataset 1 is optimised, and then validated on dataset 2, 3 and 4. All vaf values are calculated and the best average value is then used in the rest of the thesis. This will make sure that the data is not over fitted to only one dataset.

In Figure 3-5 the dataset with the highest average vaf value is plotted. The vaf Value over all datasets was 96.36% for the lateral acceleration and 98.41% for the yaw rate. The values that were the optimum are: $C_{\alpha} = 22201.3895N/rad$ for 2 wheels and $I_z = 798.8249kg/m^2$. These values were then rounded for use in simulation.

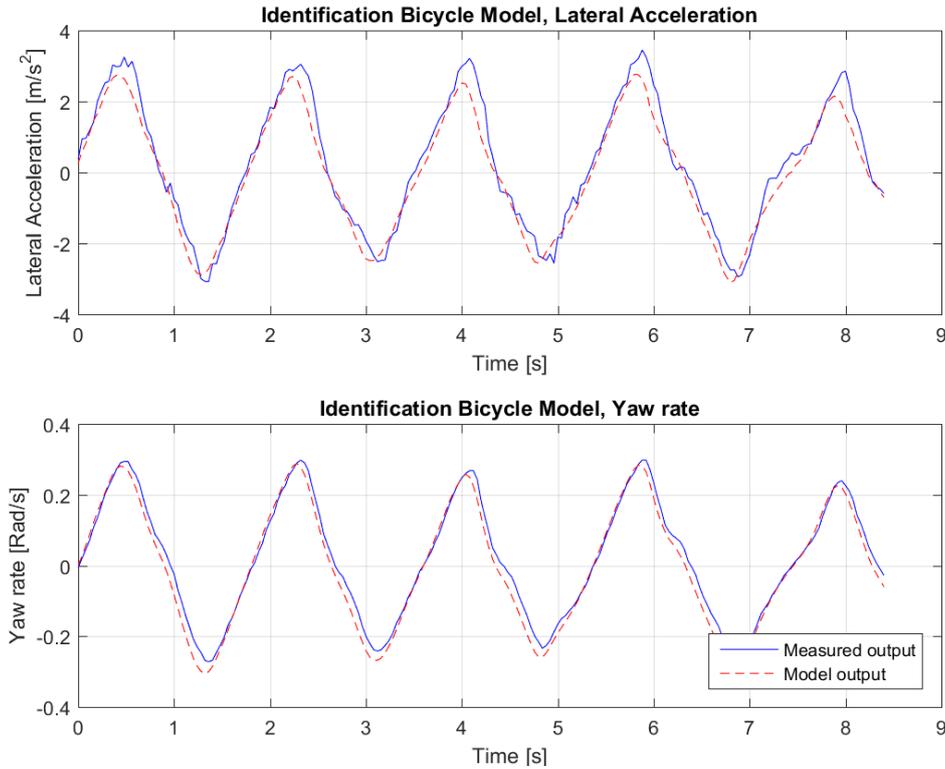


Figure 3-5: Data fit with the the highest average vaf value

3-4 Discussion

This concludes this chapter where all necessary parameters were acquired. In Table 3-3 an overview is given of the parameters that are found in this chapter. It can not be said that these parameters are the exact representation of the actual vehicle. Some may vary a little because of measurement errors or other variables. For instance the weight of the vehicle can vary because of the fuel tank that can be full or empty. And the optimized parameters may vary from reality because the system was not excited enough during the test, or it was excited to much and the tyres were going into the non-linear regime.

With all this in mind it can be said that these parameters should give a resalable good representation of the vehicle within the limits of the bicycle model.

Now that the parameters are found it is also possible to check if the vehicle model is stable. This can be done with Eq. (2-10). Because it is assumed that $C_{\alpha_f} = C_{\alpha_r} = C_{\alpha}$ it becomes clear that the vehicle is stable if $l_r > l_f$. This holds for the found parameters. Therefore the vehicle is stable.

Table 3-3: Acquired vehicle parameters

| Parameter | Value | Symbol | Unit |
|--|--------|-----------------|----------|
| Length from front axle to CoG | 1.0868 | l_f | m |
| Length from rear axle to CoG | 1.6132 | l_r | m |
| Vehicle mass | 1590 | m | kg |
| Cornering stiffness of the front wheels | 22200 | C_{α_f} | N/rad |
| Cornering stiffness of the rear wheels | 22200 | C_{α_r} | N/rad |
| Moment of inertia around Z axis | 800 | I_z | kg/m^2 |
| Steering ratio | 14.6 | k_{δ} | - |
| Maximum steering angle of the steering wheel | 7.592 | δ_{max} | rad |
| Dynamic steering time constant | 0.2 | τ_{δ} | s |

Chapter 4

Control

In this chapter the control for the path follower will be discussed. As discussed before the control will be split into two parts, a lateral controller and a longitudinal controller. This chapter will also be divided into lateral and longitudinal, and these parts will have the same build up. First it is discussed why this exact type of control is chosen and then some constraints will be introduced. Next the control algorithm itself will be described. At the end of this chapter the safety of the system will be discussed.

4-1 Controller Goals

The goal of the controllers is to make the steering and acceleration as human-like as possible. In the ultimate case this will be a perfectly human-like controller that takes the whole trajectory into consideration. This will most likely be a very advanced non-linear model predictive controller [10]. The only problem is that this controller will not be able to run in real-time because the computational load is too high. Therefore one should find a control algorithm that performs well within the limits of the system.

What if the path is changed on the go, for instance when a not yet designed path planner changes its path because there is a change of route? Or it is doing an evasive manoeuvre because a person is crossing the street? It is desired that the controller can handle this possible change and not fail to give a desired output.

The goals of the controllers are:

- **Compatible:** The controllers should be compatible with the autonomous vehicle.
- **Computational load:** The controllers must be able to run at 25 Hz in a National Instrument PXI.
- **Robustness:** The controllers should be able to handle a change in the path.
- **Comfort:** The controllers should have a response that is comfortable for the passengers.

4-2 Lateral Control

When keeping the controller goals in mind, the best solution for this problem isn't the best available controller. But it is the control algorithm that uses the available inputs, that has the best performance while it is still able to run in real-time. This results in a trade off between computational load and tracking performance.

When designing a controller one must first look at the desired output and possible inputs for this controller. For a lateral vehicle controller the output is either the steering angle of the wheels or the steering angle of the steering wheel. In the case of this theses, as described in chapter 1, it is the latter.

The inputs of a path follow system can be a large variety of signals. Some control algorithms in literature use for example the curvature of the desired path[3]. Some use the information of the path on the current position [11] [12] and others use a look-ahead distance and calculate the desired steering angle with use of this "future point" [13].

4-2-1 Comfort

Before the lateral control algorithm is introduced it first necessary to introduce the comfort restrains. An autonomous vehicle that can track a path perfectly wont be of much use when nobody wants to drive with it. Therefore the steering must feel like a human is driving the vehicle. for example a controller that has no smooth steering will result in a unpleasant drive for the people on-board of the vehicle.

The comfort level for humans can be evaluated with use of the lateral acceleration value. When this value is high the ride will not be described as pleasant by the passengers. To quantify this an experimental study [14] is used that found three threshold values of comfort, a comfortable level, a medium comfort level and a discomfort level. These are shown in Table 4-1

Table 4-1: Threshold values of comfort

| Lateral acceleration [m/s^2] | Consequence |
|----------------------------------|----------------------|
| 0 $<a_y \leq 1.8$ | Comfort level |
| 1.8 $<a_y \leq 3.6$ | Medium comfort level |
| 3.6 $<a_y \leq 5$ | Discomfort level |
| 5 $<a_y$ | Uncomfortable |

Next to this threshold it is also valid to assume that if the output of the controller is steady the comfort level will be higher. For instance when the vehicle is steering constantly left and right to maintain an straight line the comfort level of the passenger will be very low. So not only the hight of the lateral acceleration value needs to be considered but also the stability of the steering wheel output.

It should also be noted that the value of lateral acceleration is influenced by the path. If a sharp turn is made in the path at high velocity, and the vehicle is following this path than the lateral acceleration will also be high. Therefore the design of the path should also be considered when evaluating the comfort level of the controller.

4-2-2 Future Predictive Control

The controller that was selected as a result of a literature survey was the Future Predictive Controller [15]. This controller is a non-linear feedback function. The algorithm recalculates the steering angle every time-step, this means it can respond to changes in the trajectory. This control algorithm uses a future location to generate the steering angle. A control algorithm with this ability makes the most sense for this thesis because there is an ability to know the path, or at least part of it, in advance. A result of this look ahead distance is that this controller should have a better performance than control algorithms that don't use the preview of the path.

And finally one of the most important aspects of this controller is that it is proven to work real-time in literature and that is one of the goals of this thesis.

The Control algorithm is an function that exists out of two parts. One is the error between the path and the vehicles position and the other is the heading error of the vehicle with respect to the path.

The heading error can be calculated with use of equation Eq. (4-1).

$$\theta_e = \theta - \theta_p(t) \quad (4-1)$$

With θ_e being the heading error, θ is the current theta and θ_p the desired theta from the path.

In Figure 4-1 the distance error is explained. This error is not the direct error but it is the error of a projection in the future. To accomplish this projection the controller uses the vehicles current position and heading without any steering angles.

From this future point f_x, f_y a perpendicular line is drawn to the path. The length of this line is defined as the lateral error in the future y_{ef} .

The future location (f_x, f_y) can be calculated with equation 4-2.

$$\begin{aligned} f_x &= L_f \cos(\theta) + c_x \\ f_y &= L_f \sin(\theta) + c_y \end{aligned} \quad (4-2)$$

Here L_f is the prediction length ahead of the vehicle, so it is the distance between the vehicle and its prediction. The future lateral error can then be calculated with equation 4-3.

$$y_{ef} = -(f_x - p_x) \sin(\theta) + (f_y - p_y) \cos(\theta) \quad (4-3)$$

The Control law for the wheel steering is then given in equation 4-4.

$$\delta = \sin(\theta_e) + \frac{k_s y_{ef}}{v_x} \quad (4-4)$$

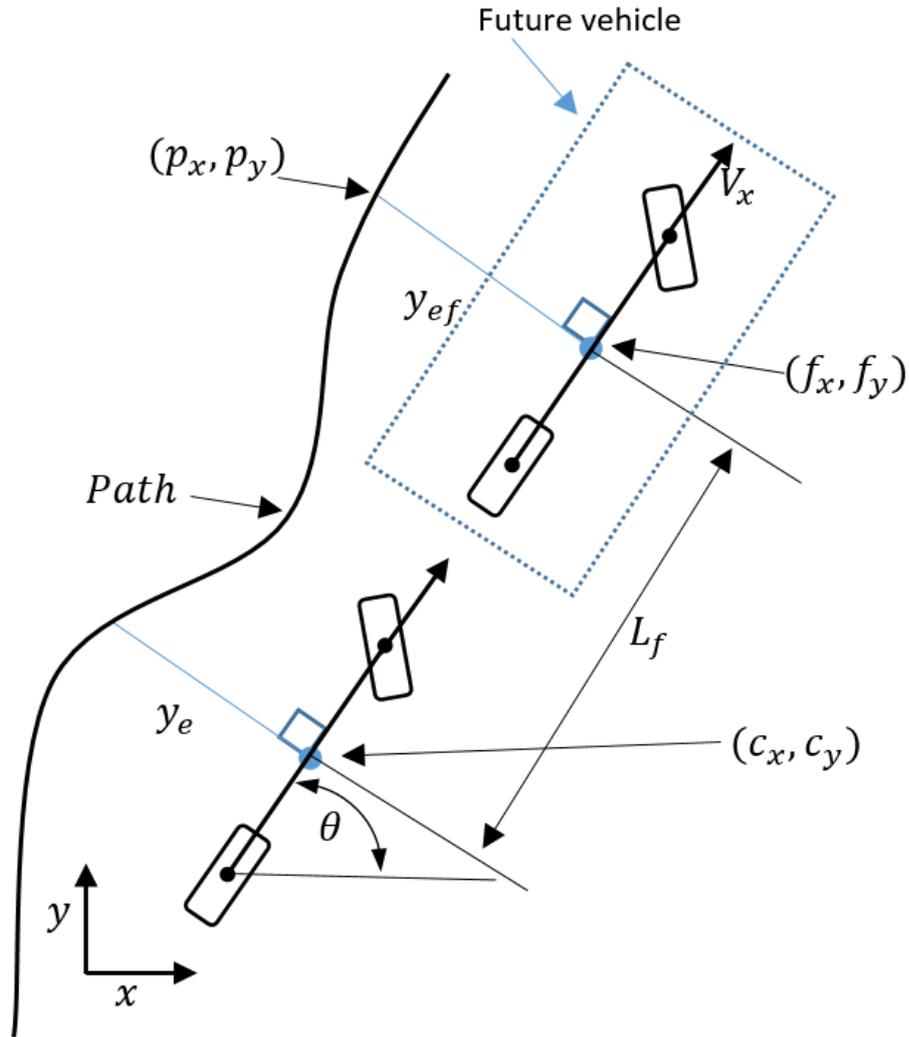


Figure 4-1: The future vehicle of the future prediction control algorithm

This control function is given in literature, but there are some improvements possible. From other literature [13] it is seen that there is an improvement in tracking if the look-ahead distance L_f is variable. This can be done by making L_f dependent of the velocity v_x . This results in:

$$L_f = K_f * v_x \quad (4-5)$$

The hypothesis is that this should have a positive effect on the overall tracking of the vehicle. Another improvement is a control parameter for the heading error. This parameter will be called k_h . This results in the control law:

$$\delta = k_h * \sin(\theta_e) + \frac{k_s y_{ef}}{v_x} \quad (4-6)$$

4-2-3 Tuning and Performance

The tuning of this control algorithm is done with three different parameters, k_f , k_s and k_h . The parameters k_h and k_s are the gain parameters. When increased the algorithm will make the controller steer more aggressively and therefore make the error decrease more quickly, this will result in a better tracking performance but can lead to an unstable system. If the control values are decreased the stability of the controller becomes better.

The other parameter that can be tuned is the k_f parameter. Increasing this parameter will result in a larger look-ahead distance. If this distance is too large the vehicle will cut corners, and if this distance is too small the vehicle will start to oscillate.

The performance of this controller should be better than similar control algorithms that lack the look-ahead distance like one shown in [16]. Only low velocity tests [25km/h] have been performed in [15], Therefore no conclusion can be drawn on the performance on high velocity. On these low velocity tests the performance is not perfectly shown in the paper. There is no figure showing the error distance but there is a figure that shows the difference between the reference and actual response. This error is measured in meters so the performance of the controller in literature is reasonable. In higher velocities the steady-state error could increase because there is no dynamic information used in the control algorithm except the velocity itself.

4-3 Longitudinal Control

This section will describe the design of the longitudinal controller. As before the comfort restrains are introduced at the beginning of this section. After that the control algorithm is explained.

4-3-1 Comfort

The lateral controller that was introduced had comfort constrains on the lateral acceleration. In this section the constrains for the longitudinal controller are introduced. These limits will be on the longitudinal acceleration. No constraints can result in very fast acceleration and deceleration even though it is not required. These high acceleration values can lead to discomfort for the passengers of the vehicle.

A statistical analysis of manual driving data of 125 drivers is presented in [17] and shows values of acceleration at different velocities. The results of this analysis are shown in table 4-2, it can be observed that 98% of the accelerations are in a range between -2.17 and 1.77 m/s^2 in low velocity situations. At higher velocities this acceleration is even less.

Table 4-2: Acceleration analysis based on velocity region

| Velocity / Percentile | 1% | 5% | 90% | 99% |
|-----------------------|-------|-------|------|------|
| 0 – 40[km/h] | -2.17 | -1.42 | 1.27 | 1.77 |
| 40 – 70[km/h] | -1.74 | -0.85 | 0.81 | 1.09 |
| Over 70[km/h] | -0.88 | -0.52 | 0.55 | 0.73 |

Most drivers and passengers feel uncomfortable when the vehicle is decelerating with values that are greater than 3 or 4 m/s^2 . If drivers decelerate with even higher values it is to prevent the vehicle from crashing into other vehicles or objects. Therefore the acceleration can be divided into 2 ranges:

$$\begin{aligned} \text{Normal driving: } & -2.17[m/s^2] \leq a_c \leq 1.77[m/s^2] \\ \text{Severe braking: } & -4[m/s^2] \leq a_c \end{aligned}$$

For normal driving conditions the constraints can be added to the control law as:

$$a(t) = \begin{cases} a_{max}(v_c(t)) & \text{if } a_c(t) > a_{max}(v_c(t)) \\ a_c(t) & \text{if } a_{min}(v_c(t)) \leq a_c(t) \leq a_{max}(v_c(t)) \\ a_{min}(v_c(t)) & \text{if } a_c(t) < a_{min}(v_c(t)) \end{cases} \quad (4-7)$$

The maximum and minimum accelerations are depending on the vehicles velocity and are determined with use off the human driving data. These acceleration values are shown in figure 4-2.

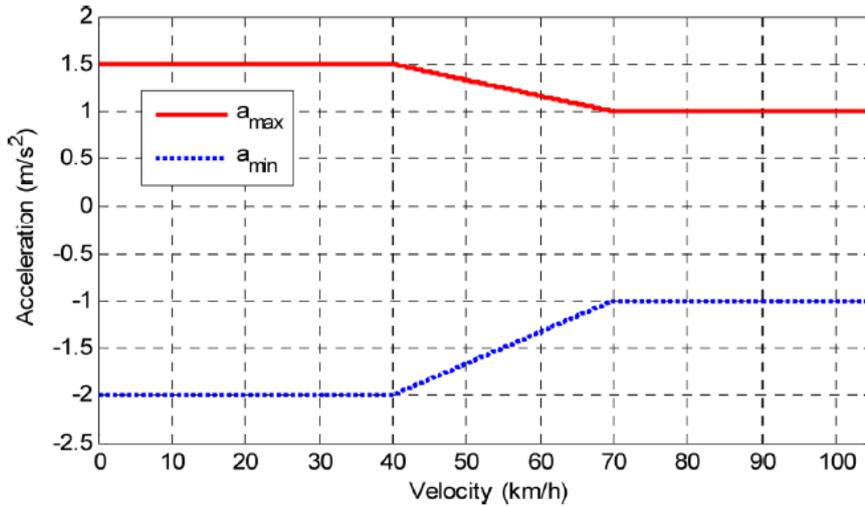


Figure 4-2: Velocity-dependent rang of accelerations

4-3-2 Safety

Limiting the acceleration value of braking doesn't sound very safe, especially when a situation occurs where an emergency brake is necessary. This is completely true and for that reason there are more advanced longitudinal controllers available in literature [18]. For instance a controller that uses a Time To Collision (TTC) and a warning-index to compute the necessary deceleration value. In this thesis however this controller is not used because the tracking systems of the Toyota Prius are not yet fully functioning and the research of this thesis is on the tracking of the path without any disturbances of other road users. The real test will not be done on open road but on a closed controlled track environment and therefore such safety systems will not be required.

4-3-3 PD Control

The longitudinal controller should, as discussed in chapter 1, have as output the desired acceleration of the body. The desired velocity will be part of the path that is constructed or given from a path planner. Because there is already an Adaptive Cruise Control (ACC) controller inside the Move-Box this longitudinal controller doesn't have to be very advanced. The ACC has the most influence on the performance but unfortunately little is known of the controller because it is part of the standard equipment of the Toyota Prius and Toyota doesn't give information about these systems.

The longitudinal control part of the path follower, will follow the desired velocity that is given in the path. To control the velocity a proportional derivative or PD controller can be constructed to regulate the acceleration of the vehicle. To do so it is desired to know the velocity error, this can be calculated with equation 4-8.

$$v_e(t) = v_c(t) - v_p(t) \quad (4-8)$$

Where $v_e(t)$ is the error in velocity on time t , v_c is the vehicles velocity and v_p is the velocity that is requested by the path. The acceleration control law can then be defined as:

$$a_c(t) = K_p v_e(t) + K_d \frac{dv_e(t)}{dt} \quad (4-9)$$

The parameters of this control law K_p and K_d are the proportional control gain and the derivative control gain. These parameters can be tuned to make the vehicles velocity converge quickly to the desired velocity.

4-4 Discussion

In this chapter the lateral and longitudinal controller are introduced. First the requirements for the controllers were discussed. The control algorithms were selected on there real-time capabilities and the robustness of control. This was done while keeping in mind the limits of the set-up, the Toyota Prius.

For both controllers comfort restrains were introduced and explained. Then the control algorithm themselves were introduced, and the tuning of the lateral controller was examined

In the next chapter a system will be designed that will acquire the necessary parameters for the lateral and longitudinal control from the real-time data that is available of the Prius. Also the design of the path will be introduced.

Chapter 5

System Design

In this chapter the design of the path following system is introduced. The chapter starts with explaining what program is used throughout the thesis and why this program is chosen. Then a key input of the path follower is discussed, the path itself. The design of this path and the requirements will be introduced. The rest of this chapter will then discuss the system that is designed to make sure the controllers of chapter 4 get the required data from the vehicle.

5-1 LabVIEW

The program used to construct the system is LabVIEW from National Instruments. This is done because one key goal of this thesis is the real-time implementation of the path follower. LabVIEW has the possibility to construct a Virtual Instrument (VI). With these VIs it is possible to construct models that can be simulated when the time is synchronised to a timing source. Thus making it possible to have real-time simulations.

This are capabilities that are in a similar way also possible in other programs like Matlab. The only difference is that National Instruments isn't only software. They have a variety of hardware that is compatible with the software of LabVIEW. The hardware is build based on the LabVIEW system and therefore there is no problem with drivers. This means that a code programmed in LabVIEW can easily be compiled to such device from which it is possible to run the code in real-life systems.

The Move-Box in the Toyota Prius is actuated by a PCI eXtension for Instrumentation (PXI) from National Instruments. Therefore the choice to program the system in LabVIEW is logic. The implementation of the system to the PXI can then be done without much trouble, and then it is possible to test if the full system is able to run real-time. When processed on a Windows PC you do not achieve the maximum process power because a lot of other programs are running.

5-2 The Path

The most important input of a path following system is the path it needs to follow. This essential path is already mentioned in the other chapters but what does it really mean? In this section an answer to that question is given.

This trajectory exists out of a long list with data points. One row of this list correspond to one point in the map as shown in figure 5-1. Therefore the list will contain a column of x points and a column of y points. Next to that it does also contain the heading angle θ the desired velocity v_x and the value L_p that denotes the length of the path from the starting point. This list is shown in Table 5-1.

Table 5-1: Data of the path represented in a table

| | Position x | Position y | Heading angle | Velocity | Length of the path |
|----------|------------|------------|---------------|-----------|--------------------|
| p_1 | x_1 | y_1 | θ_1 | v_{x_1} | L_{p_1} |
| p_2 | x_2 | y_2 | θ_2 | v_{x_2} | L_{p_2} |
| \vdots | \vdots | \vdots | \vdots | \vdots | \vdots |
| p_N | x_N | y_N | θ_N | v_{x_N} | L_{p_N} |

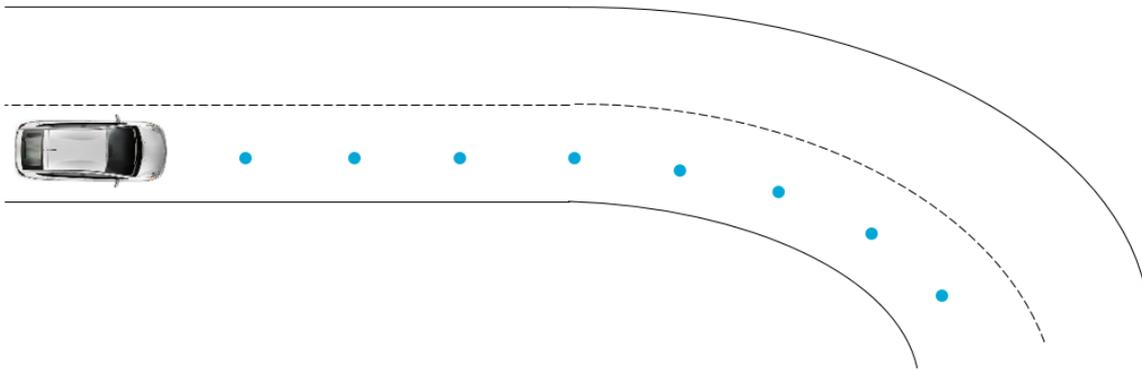


Figure 5-1: A visualisation of the path

The position values x and y represent a point in a global coordinate system. The system chosen is the Universal Transverse Mercator coordinate system (UTM). This coordinate system divides the world into rectangles. The benefit of this system is that it is defined in meters and therefore it can be used to calculate distances.

The heading angle will be in radians. It will be a value between 0 and 2π that is increasing counter clockwise from a polar axis that is drawn horizontal and pointed to the right. In this way trigonometrical equations will get the expected results.

The velocity will be in meters per second and although in theory it should be possible to track a path in reverse in this thesis a negative velocity is not investigated.

5-2-1 Generation of a Trajectory

This trajectory can be generated in different ways. It can be computed by using different lines or function connected together, or it can be recorded with the actual vehicle. When recording with the vehicle it is required to have a good satellite reception.

The number of points of the trajectory depends on the length of the path, but the distance between the points will be set constant. To be able to have a good functioning system the distance between every point will be set to $0.05m$. This means that lines or an actual driven path need to be interpolated. This interpolation is done in Matlab with the function `interparc`. To calculate the heading of a path that is generated from lines the function `myfrenet` is used.

When interpolating the path one should also interpolate the heading. One should consider that the heading line is not a continuous curve. If the heading should reach an angle of 2π it returns to zero. When this line is interpolated there will be points from 2π back to zero that will disturb the controller. Therefore it is necessary to first create a continuous line of the heading, interpolate it and then write it back between 0 and 2π as is shown in Figure 5-2.

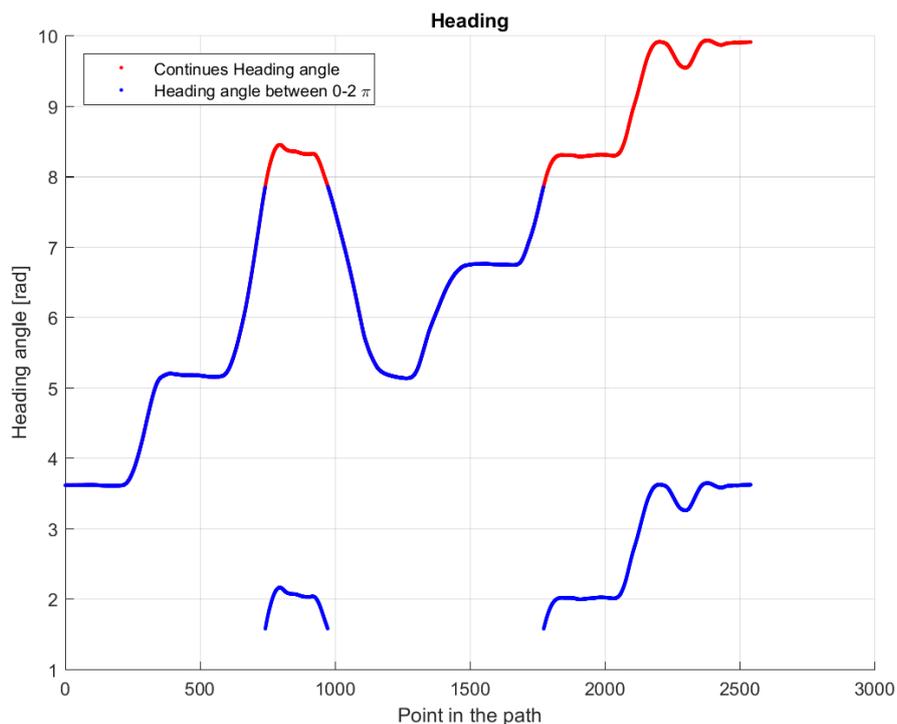


Figure 5-2: Creating a continuous line for the heading

The heading angle of the path that is measured from the Global Positioning System (GPS) can also be used in the path. The same interpolating action as described above need to be applied. In the trajectories that were generated in this thesis there was a difference between the calculated heading angle and the measured angle. This is shown in Figure 5-3. In the upcoming chapter 7 it will be shown that using the GPS measured path will result in more unstable tracking with worse results. This is probably due to the fact that the paths were recorded at very low velocity and, as was later discovered, when driving very slowly or

even standing still the GPS heading will "jump around". This can be an explanation to the difference between the heading angle from GPS and the calculated one. It is also possible that at the day of recording the connection with the satellites was not stable enough.

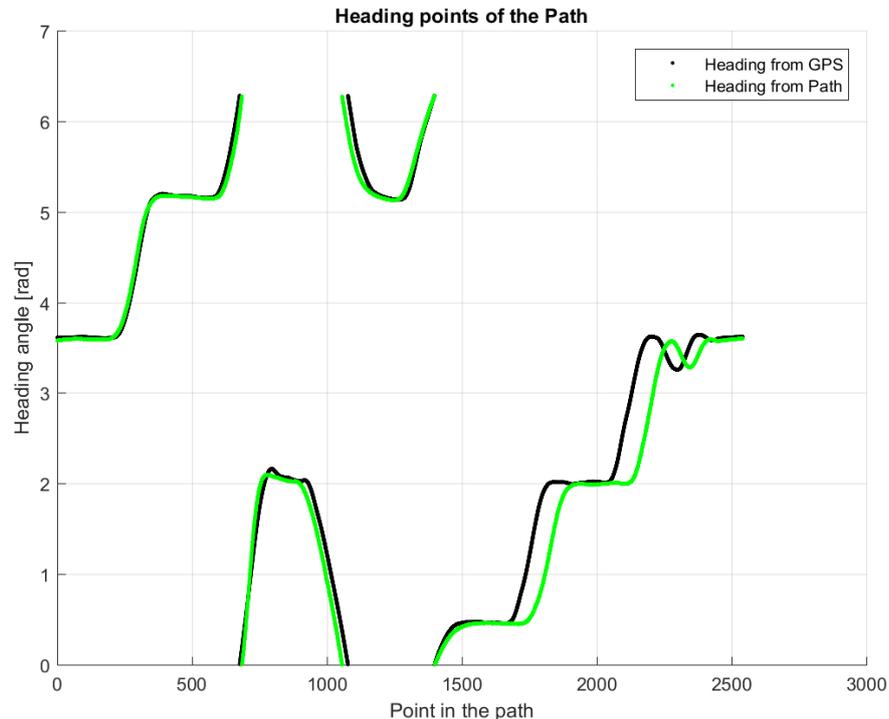


Figure 5-3: the heading calculated from GPS and the heading calculated from the path

5-3 The System

In this section the path follower will be described. Although the path follower does include the controllers of chapter 4 it does consist out of more parts than only the controllers. That is because it is necessary to generate the desired inputs for the controller. For the longitudinal controller this means finding the desired velocity and feeding the current velocity.

The lateral controller does require some more advanced inputs, it needs the perpendicular distance to the path of a projection of the vehicle, the heading error and the current velocity.

The system that is designed to generate these outputs is shown in Figure 5-4. It exists out of 5 subsystems and a vehicle model. In the simulations the vehicle will be replaced by the vehicle model. The 5 blocks are:

- LLtoUTM, this subsystem takes care of the conversion from latitude and longitude to UTM coordinates [19].
- Curve Equation, this subsystem finds the nearest path point and makes a curve fit to a part of the trajectory.
- Calculations, the subsystem will calculate the distance between the path and the vehicle.

- The Longitudinal controller.
- The Lateral Controller.

A schematic view of this system can be seen in Figure 5-4.

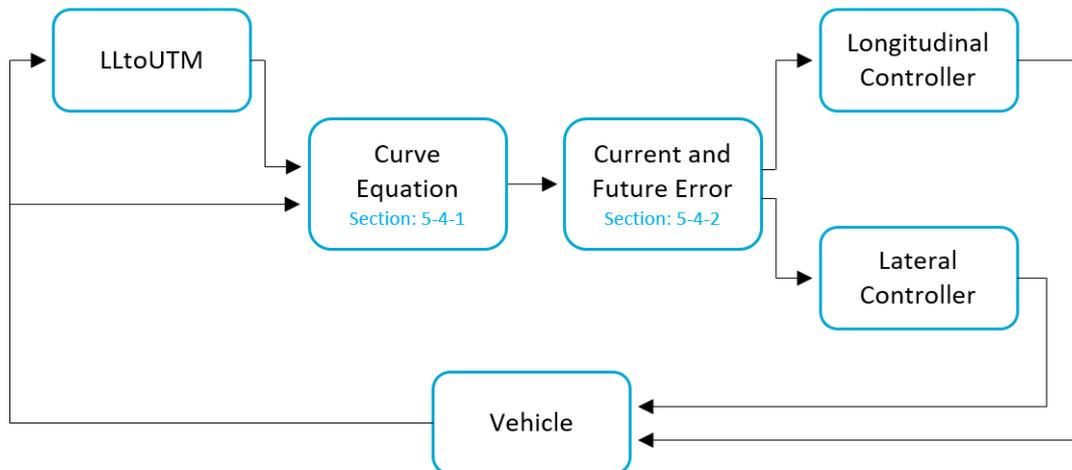


Figure 5-4: Schematic view of the path follower

5-4 Perpendicular Distance to The Path

The controller requires the perpendicular distance to the path calculated from the centre of the vehicle. This means it is not the closed line between the vehicle and the path, because this could be a line that is not perpendicular. This concept is shown in figure 5-5. The dotted red line is the shortest line from the vehicle to the path. The blue line is the perpendicular line from the vehicle to the path denoted with y_e , the error in lateral position.

To calculate the distance y_e it is necessary to have 2 points: the current vehicle position (c_x, c_y) and a point on the line of the path that is perpendicular to the current vehicle position (p_x, p_y) . The location of the vehicle is available with use of GPS but the point on the path is more difficult.

This problem will be solved by using 2 line equations. One will be a curve fitted to a small part of the path. The other line will be created by using the heading angle θ of the vehicle. A line is created from the centre of the vehicle with a slope equal to the heading angle plus 90 degrees. This creates a line perpendicular to the vehicle with a very simple equation, $y = ax + b$.

With these two lines the perpendicular point can be found by determining the intersection point of the lines. This results in the point (p_x, p_y) . The distance y_e is now the difference between (c_x, c_y) and (p_x, p_y) . This same idea can be used to calculate the perpendicular distance of the projected vehicle. Only one extra step is required and that is adding the length L_f in the direction of the vehicle heading θ to get the point c_{x_f}, c_{y_f} .

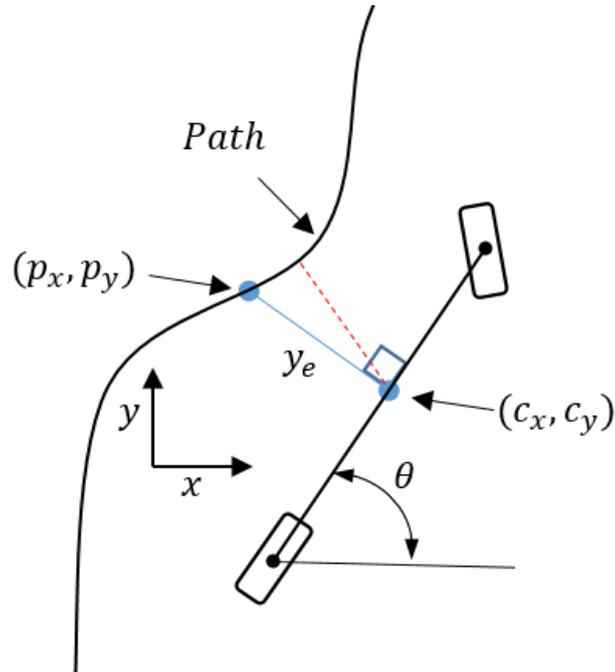


Figure 5-5: The bicycle model and the perpendicular distance to the path

Both of these two distances will be used in the rest of the system. y_{ef} will be an input to the controller and y_e will be used to evaluate the performance.

5-4-1 Curve Equation

This subsection begins with rewriting the θ of the vehicle, θ_c , to the desired situation. This is necessary because trigonometrical functions will be used later in the system. The heading of the vehicle calculated with the GPS system results in a heading between 0 and 2π that is increasing clockwise from a polar axis that is drawn vertically and pointed up. This difference is shown in Figure 5-6. To rewrite this θ_c Eq. (5-1) is used.

$$\begin{aligned} \theta_c &= |\theta_c - 2\pi| + \frac{1}{2}\pi \\ &\text{if } \theta_c > 2\pi \\ &\quad \theta_c = \theta_c - 2\pi \\ &\text{end} \end{aligned} \quad (5-1)$$

Nearest Path Point

After this conversion the system will find the nearest path point. This is done with use of the last given path point, or if it is the first iteration, a starting point. A pre-generated path can vary in length and therefore it can become a huge list of data points. Finding the nearest path point in that list will increase the computational load of this system. Therefore a small section of the path is evaluated. The last point P_n will be the beginning of the search, and

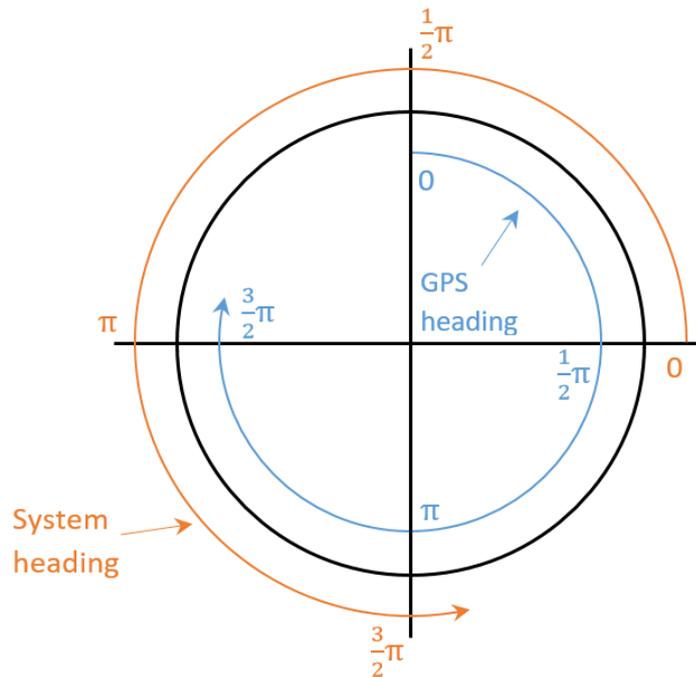


Figure 5-6: The heading error θ_c before and after conversion

the size of this search area will depend on the current velocity. When the vehicle is driving faster the number of path points will increase because more points are skipped. In that way it will always be possible to find an accurate closed point on the path without having to search through the whole trajectory list.

One benefit of this is that it will be possible for the trajectory to cross itself, if these crossing points have enough space between them.

When the nearest point is selected a section of the path around this point will be extracted from the trajectory list, a few data points backwards on the path and a large amount of data points forward on the path. This number will again be defined depending on the forward velocity and a control parameter k_{path} . This value k_{path} needs to be larger than the control parameter k_f of chapter 4. Otherwise the system will not be able to calculate the distance y_{ef} .

The required velocity v_x and required heading angle θ are selected from the same line as the closed point P_n .

Curve Fit

Now that a part of the trajectory is selected it is possible to fit a curve over these data points. There is only one problem left to handle. The curve that will be fit is a polynomial curve, that will result in a suitable equation form which it is possible to calculate the intersection. The only thing with curve fitting a polynomial to data is that the data on the x-axis should be only increasing. It is not possible to fit a polynomial curve over a line that is moving

Although this seems to be the solution to the previously mentioned problem, there is a bit of this problem left. Namely when the trajectory will make a u-turn or a sharp corner it may occur that even after converting the trajectory from global to local the x axis data is still moving backwards in X. For instance the path shown in Figure 5-8a. This is a local path of a u turn. When this happens the fit of this curve will sometimes fail, or it will have a large deviation from the path.

A solution for this problem is to rotate the figure, as is shown in Figure 5-8b. One can now see that the data on the Y axis is only increasing.

The solution is to fit the data twice, one time with the X data as the X data and the Y data as the Y data, and the second time the rotated figure. So the data of X will become Y and vice versa. Choosing the best fit will then be done by choosing the fit with the lowest mean squared error.

This error doesn't occur a lot, because these sharp turns are normally made with low velocity, and the length of the path section is dependent of the velocity. Nevertheless with this solution it will never appear.

A nice benefit of this solution is that there are always 2 fits to chose from. For instance when the data is generated from a GPS path the data will not be in a perfect line. Then a double fit will make sure there is redundancy in the system.

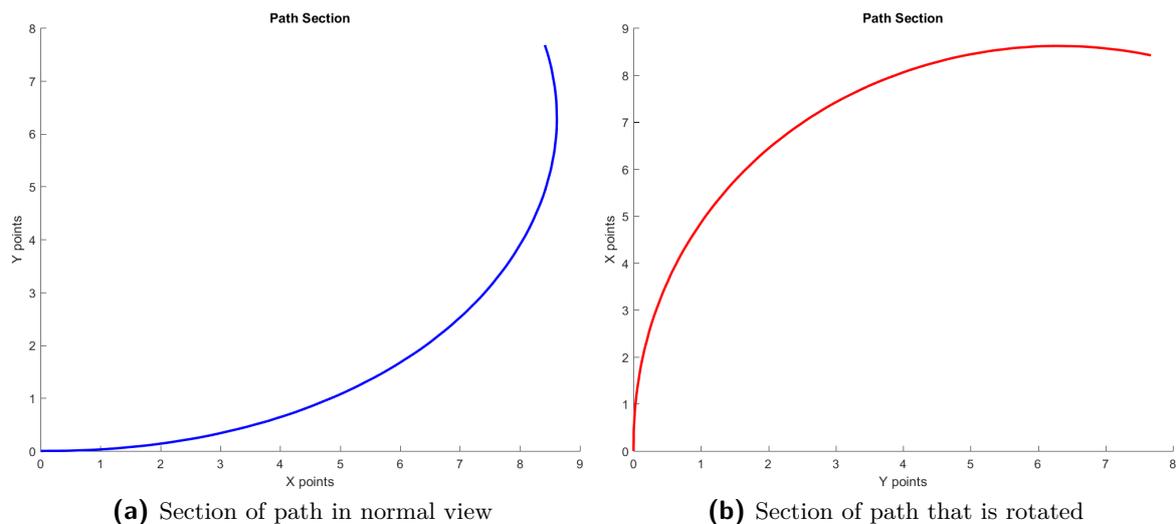


Figure 5-8: The same path section plotted in different ways

5-4-2 Current en Future Error

The next subsystem will receive the fitted curve and the current values of the vehicle. With these parameters the current and future position are calculated. From these points lines are created that are perpendicular to the heading of the vehicle. The intersection points of these lines with the path curve are found by using the `Polynomial Roots VI`. With the found points it is possible to calculate the distances y_e and y_{e_f} . This is shown in Figure 5-9

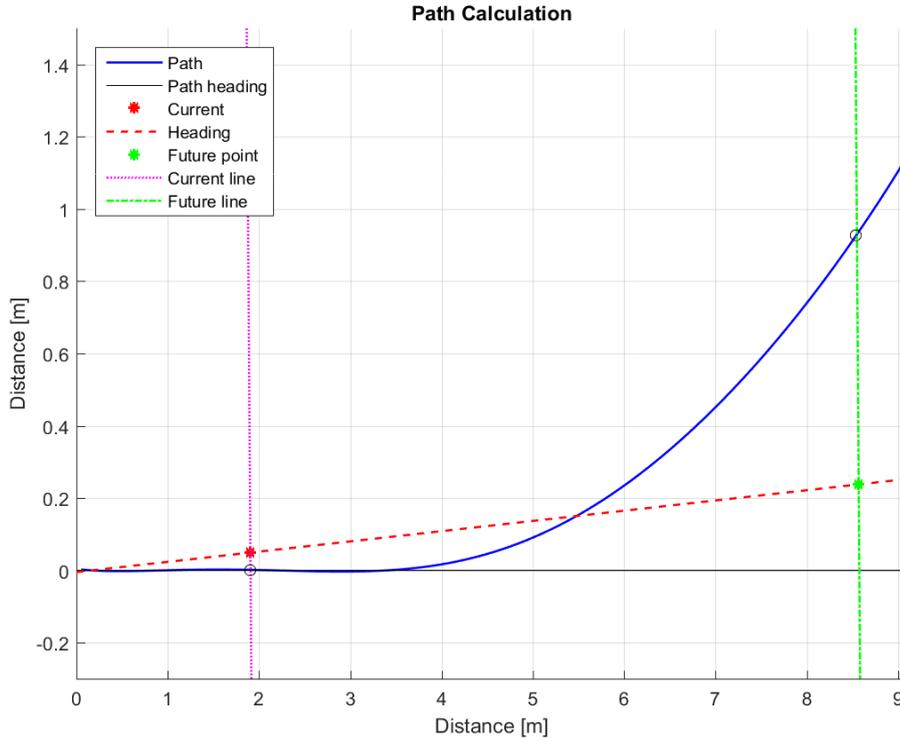


Figure 5-9: Schematic explanation of the path follower

Besides the lateral distances the value of θ_e is calculated. In some cases it can occur that this error is very large, this happens when either the path or the vehicle has passed the 2π and the other parameter has not. Then this error becomes almost 2π or more. To solve this a small loop is created that checks if the error becomes larger than π . This can be seen in Eq. (5-3).

$$\begin{aligned}
 &\text{if } \theta_e > \pi \\
 &\quad \theta_e = \theta_e - 2\pi \\
 &\text{elseif } \theta_e < -\pi \\
 &\quad \theta_e = \theta_e + 2\pi \\
 &\text{end}
 \end{aligned} \tag{5-3}$$

5-4-3 Local to Global Position

The model of chapter 2 will be implemented for simulation. The lateral model has two outputs v_y and r , and the longitudinal model gives the current velocity V_c . The inputs of this system, as previously described, are the current position X_c and Y_c together with θ_c and V_c . Therefore it is necessary to calculate these values from the outputs of the model.

θ_c can be calculated by integrating r . The movement of the vehicle will be calculated with Eq. (5-4). Then the current position can then be calculated by integrating \dot{X} and \dot{Y} ,

$$\begin{aligned}
 \dot{X} &= -v_y \sin(\theta_c) + v_x \cos(\theta_c) \\
 \dot{Y} &= -v_y \cos(\theta_c) + v_x \sin(\theta_c)
 \end{aligned} \tag{5-4}$$

Chapter 6

Simulations

This chapter will start by describing the paths that will be used for simulating and tuning the controller. Then the tuning process will be described. The performance of the controller will be evaluated, and the previously introduced improvements will be reviewed.

6-1 Testbed Paths

To understand the tuning of the controller it is required to have a path that makes the different aspects stand out. Therefore it is not a good idea to start with a large trajectory with different corners, because it will not be clear what changes of the control parameters have a positive influence and what have a negative influence.

For the tuning of the control a curve of constant radius will be used. The longitudinal velocity will be taken as constant to have no disturbance from accelerating and decelerating. Following a trajectory with different velocities will be part of the next chapter.

A suitable curve that can be followed at constant velocity is found by the use of literature. In [1] a minimum horizontal curve radius for highways is given depending on the maximum speed. This data is constructed with use of the specification of the German RAL guide that states that the critical absolute value of lateral acceleration is $1.47[m/s^2]$ for highways. The minimum horizontal curve values are shown in Table 6-1. From this table the trajectories are constructed. A curve of $1/4$ of a circle is created with a straight line before and after the curve. The length of this straight line depends on the velocity, for example if the velocity is $30km/h$ then the length will be 30 meters.

Table 6-1: Minimum horizontal curve radius for constant velocity

| Velocity [km/h] | Minimum Radius [m] |
|-----------------|--------------------|
| 20 | 15 |
| 30 | 35 |
| 40 | 55 |
| 50 | 85 |
| 60 | 125 |
| 70 | 170 |
| 80 | 220 |
| 90 | 280 |
| 100 | 345 |
| 110 | 415 |
| 120 | 495 |
| 130 | 580 |

6-2 Tuning

Tuning the controller is not a simple task. There are three parameters that influence the performance, the steering parameter k_s the look-ahead distance parameter k_f and the heading parameter k_h . To determine the influence of these parameters, one parameter will be changed and the other two will be held constant. The first evaluations will be done on a curve with a radius of 55 meters and with a constant velocity of 30 kilometres an hour. All the simulations of this chapter will be done with a frequency of 100 Hertz.

In the next two subsections the influence of two tuning parameters are shown. The heading parameter k_h will be kept constant as 1. After some tuning the best combination of parameters is found as $k_s = 0.7$ and $k_f = 1.1$. These values, and the corresponding response, will be the base for explaining the influence of the parameters.

Evaluating the performance of the controller will be done by looking at the response of the controller, the steering wheel output, the error distance y_e and the lateral acceleration a_y . But difference between some parameters will be difficult to spot. Therefore for every simulation a table with evaluation parameters is added. In this table the following parameters are shown:

- The Root Mean Square (RMS) of the error distance y_e .
- The maximum error distance y_e .
- The minimum error distance y_e .
- The total error distance y_e , a sum of the absolute values.
- The maximum lateral acceleration value a_y .

6-2-1 Steering Parameter k_s

To evaluate the steering parameter k_s three different simulations are plotted in figures 6-1, 6-2, 6-3 and 6-4. Before these simulations are analysed it is necessary to explain the different values that are shown. In the figures all values can have a positive value and a negative value. For the steering angle θ it holds that a positive angle is steering left, a negative angle is steering right. The distance errors y_e and y_{ef} are positive if the error is measured from the left-side to the path, and they are negative if it is measured from the right-side. The heading angle error θ_e is positive if the heading angle of the vehicle is larger than the required angle. This means that the vehicle is facing more to the left than required. if θ_e is negative than it is the opposite way.

- Sim1 is with the values that are previously described, $k_s = 0.7$ and $k_f = 1.1$. These are the parameters that generate the best response.
- Sim2 has a smaller steering parameter: $k_s = 0.1$.
 - When the value of k_s is too low the vehicle will have a slower response to the corner. The distance error will increase until the end of the corner. This happens because the steering output is more based on the heading angle than on the distance y_{ef} .
 - It can be observed that θ_e becomes smaller and this results in less output of the steering angle, even-though y_{ef} is increasing.
 - y_e stays large even after the corner. The vehicle is more steered with θ_e and this steering parameter wants the vehicle to be horizontal. So the steering parameters are contradicting each other.
- Sim3 has a higher steering parameter $k_s = 1.3$.
 - the controller reacts harder on the future error distance y_{ef} and therefore this value is smaller than of the other simulations. As a result the error distance y_e becomes positive, and the vehicle will cut the corner.
 - This can also be seen in the response of the heading error θ_e . That error becomes larger in the first seconds of the corner, what implies that the vehicle is more steered to the left than required.
 - The steering angle of the third simulation takes more time to settle, this seems to be a result of the larger heading error that takes more time to reach back to zero.

Table 6-2: Evaluation parameters for variation k_s

| | | Simulation 1 | Simulation 2 | Simulation 3 |
|--------------------|---------------------|--------------|--------------|--------------|
| Steering Parameter | k_s | 0.7 | 0.1 | 1.3 |
| RMS | [m] | 0.052 | 2.044 | 0.331 |
| Max y_e | [m] | 0.097 | 0.010 | 0.601 |
| Min y_e | [m] | -0.099 | -3.220 | -0.001 |
| Sum of $ y_e $ | [m] | 55.458 | 2283.916 | 354.492 |
| Max a_y | [m/s ²] | 2.282 | 2.165 | 2.188 |

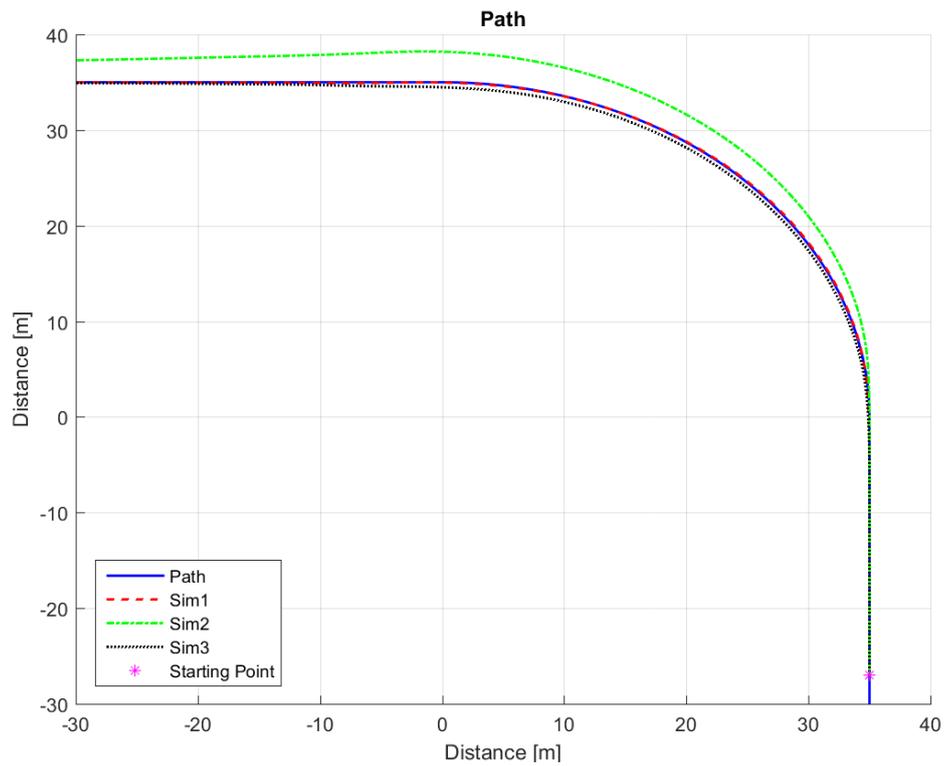


Figure 6-1: The path and simulation responses of variation k_s

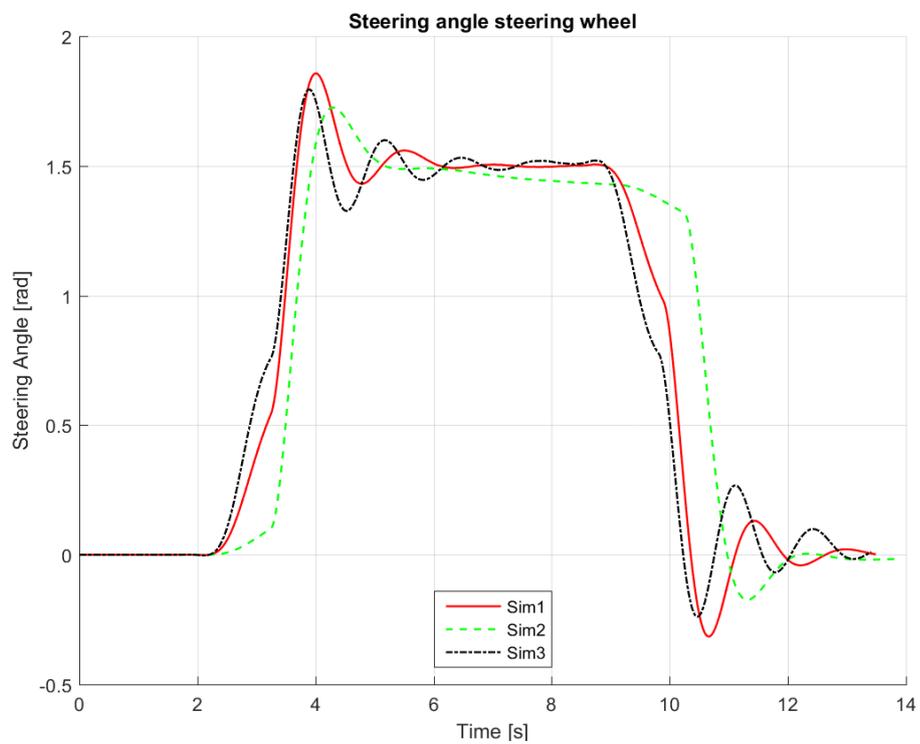


Figure 6-2: Steering angle responses of variation k_s

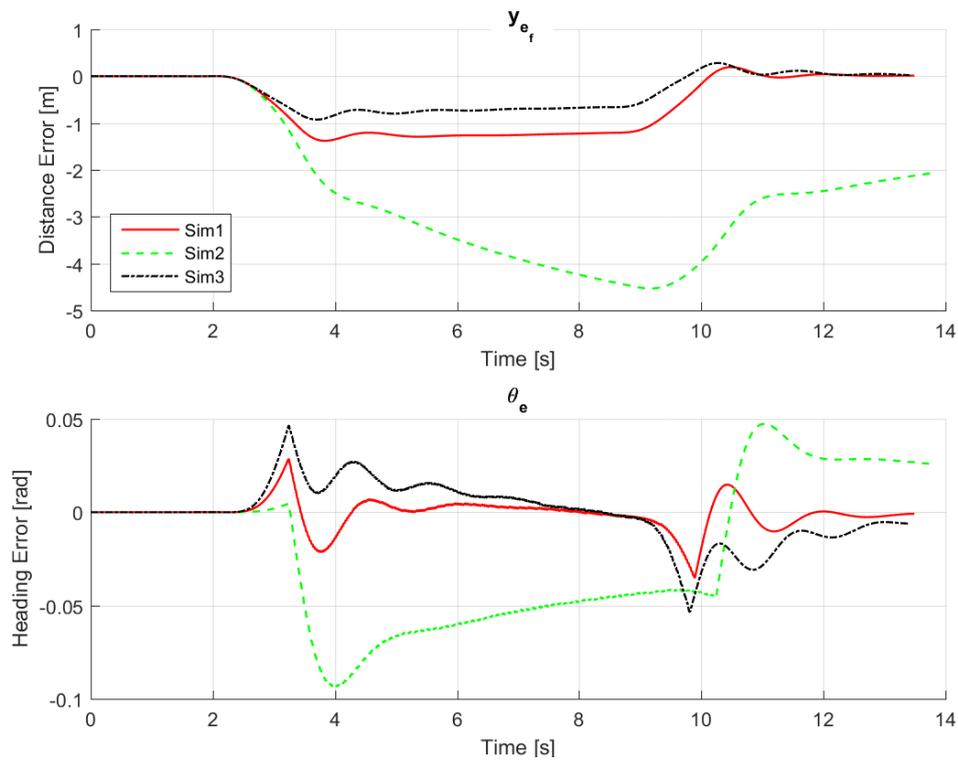


Figure 6-3: Control input for different simulations of variation k_s

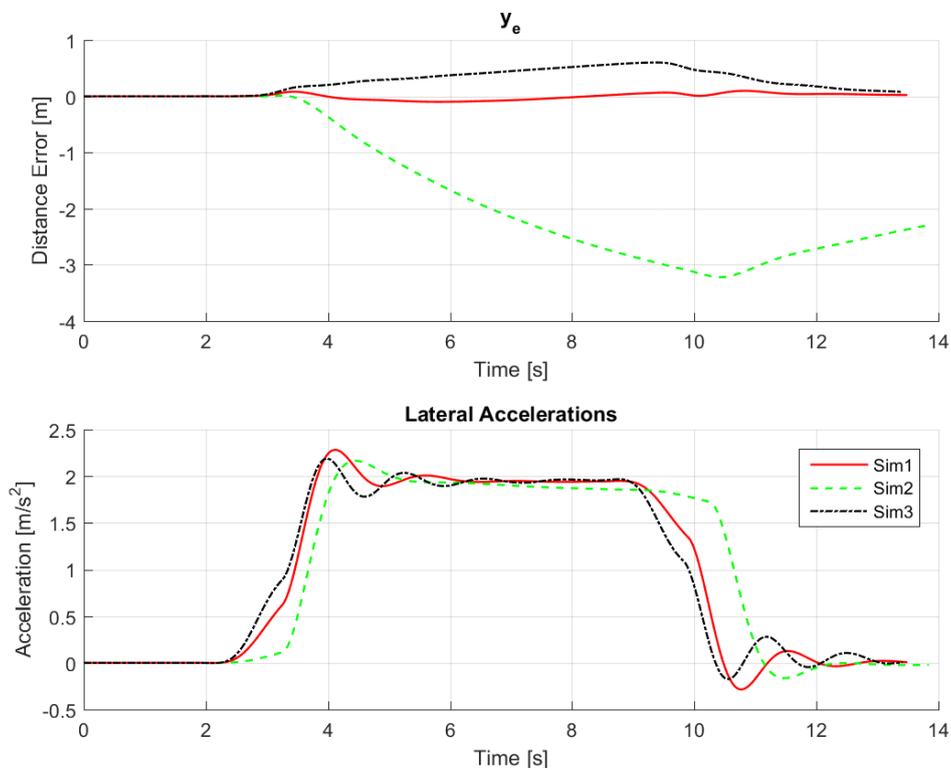


Figure 6-4: Evaluation data of different simulations of variation k_s

6-2-2 Look-Ahead Distance Parameter k_f

To evaluate the steering parameter k_f three different simulations are plotted in figures 6-5, 6-6, 6-7 and 6-8.

The difference with changing k_s can directly be observed. The larger error doesn't come from the smaller control parameter, as was the case with the change of k_s . This time the larger error occurs when the parameter is larger than the optimum.

Next to differences with the change of k_s there is also a resemblance, again a larger value results in cutting corners and a smaller value in a slower steering response.

- Sim1 is with the values that are previously described, $k_s = 0.7$ and $k_f = 1.1$. These are the parameters that generate the best response.
- Sim4 has a smaller look-ahead distance parameter: $k_f = 0.1$
 - The steering response is more influenced by the heading error than by the future distance error. These errors start to increase roughly at the same time because there is little look-ahead distance. The result is overshoot.
 - The response of y_{e_f} is almost the same as in Sim1, the only thing that is different is that it happens later, so decreasing this distance parameter will make the vehicle respond later to changes in the path, and therefore the heading error will become larger.
- Sim5 has larger look-ahead distance parameter: $k_f = 2.1$.
 - The look-ahead distance is so large that the vehicle starts steering way before the corner is actually there. Then when the vehicle is turned slightly, the look-ahead distance will cross the path, making the calculations in such a way that the error distance y_{e_f} becomes positive again, making the vehicle turn to the right. This will start a oscillating reaction that damps out after a while, only to begin again at the end of the corner.

Another observation is that the acceleration values of both simulation 4 and 5 are higher than simulation 1. When changing the value of k_s they both became smaller. This can be explained by looking at the steering angle in Figure 6-6. both of the steering angles of Sim4 and Sim5 are, at some point, larger then the steering angle of Sim1.

Table 6-3: Evaluation parameters for variation k_f

| Look-Ahead Parameter | k_f | Simulation 1 | Simulation 4 | Simulation 5 |
|----------------------|-----------|--------------|--------------|--------------|
| | | 1.1 | 0.1 | 2.1 |
| RMS | $[m]$ | 0.052 | 0.769 | 1.633 |
| Max y_e | $[m]$ | 0.097 | 0.002 | 2.708 |
| Min y_e | $[m]$ | -0.099 | -1.184 | -0.0001 |
| Sum of $ y_e $ | $[m]$ | 55.458 | 794.082 | 1783.990 |
| Max a_y | $[m/s^2]$ | 2.282 | 2.623 | 3.624 |

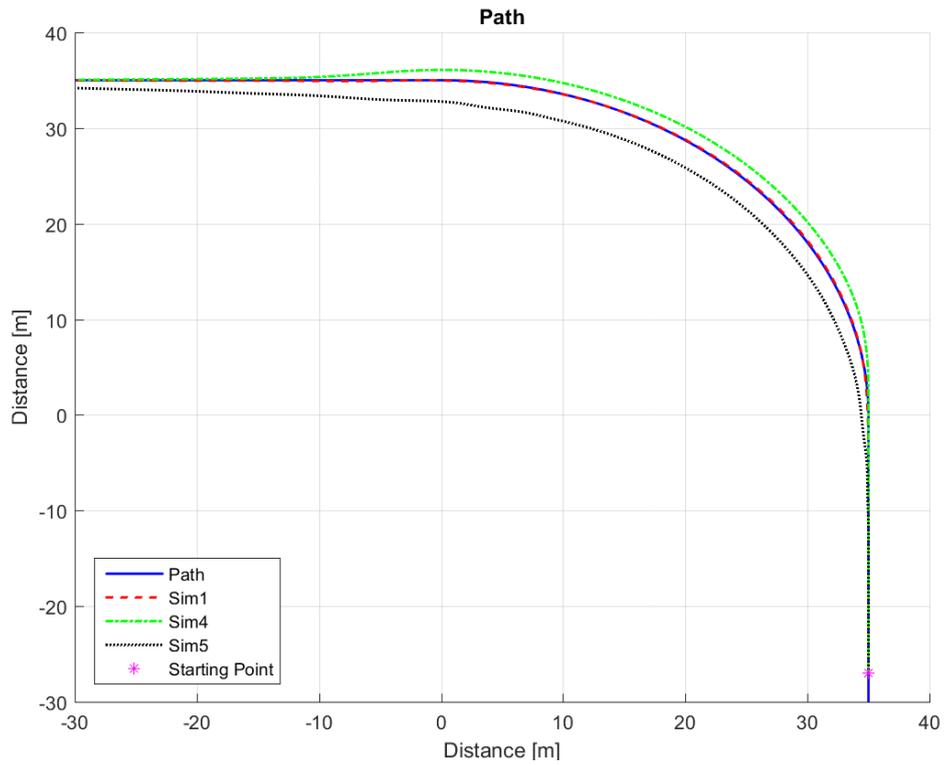


Figure 6-5: The path and simulation responses of variation k_f

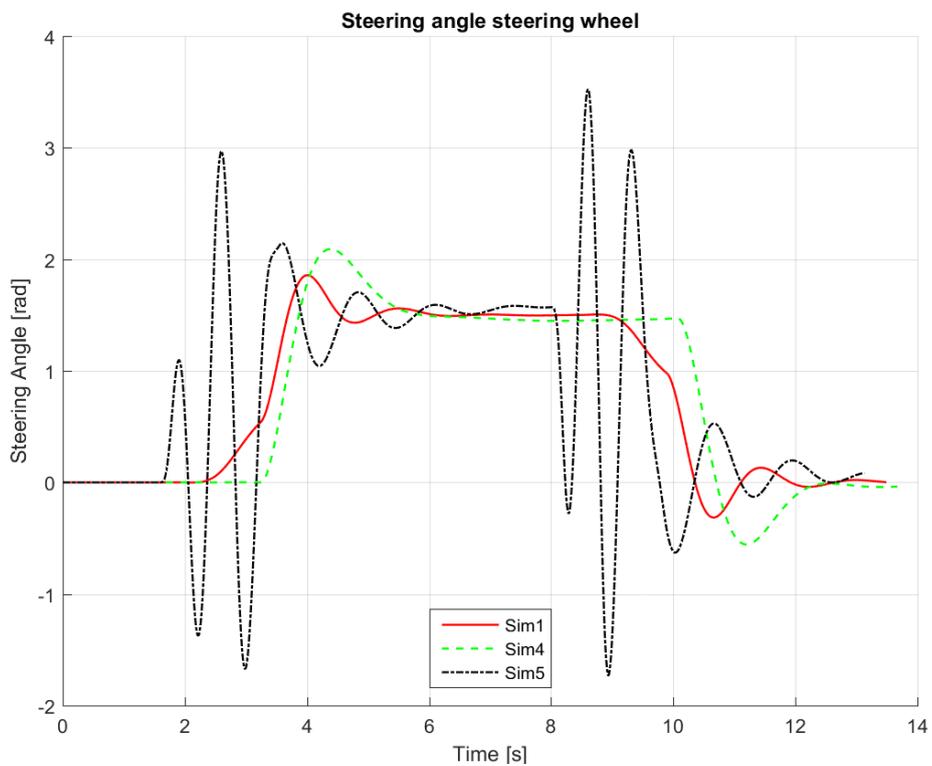


Figure 6-6: Steering angle responses of variation k_f

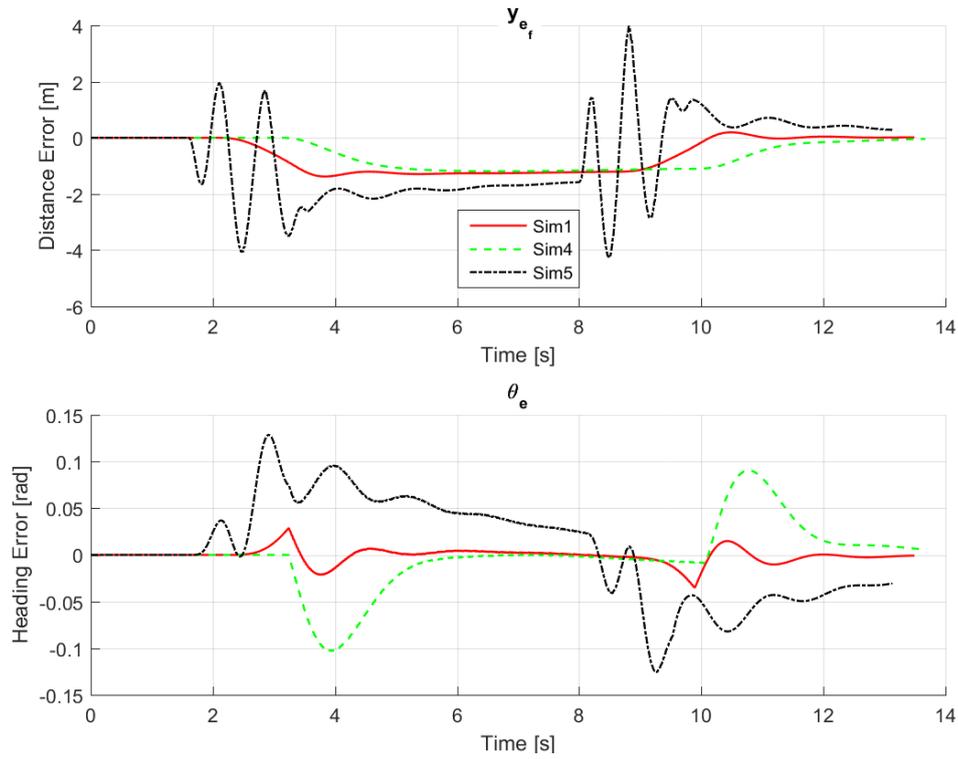


Figure 6-7: Control input for different simulations of variation k_f

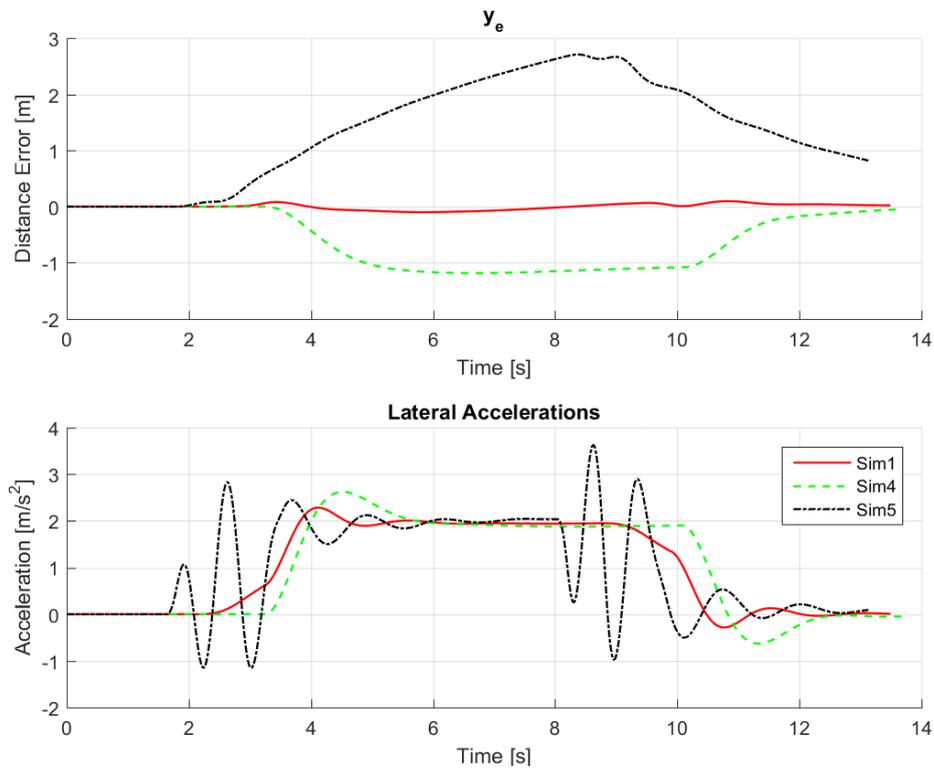


Figure 6-8: Evaluation data of different simulations of variation k_f

6-2-3 Steering Output

Another result of these simulations is that the steering output can be evaluated. There is unwanted response in the steering angle. This is not really visible in the previous figures. Therefore the Simulation with $k_s = 0.7$ and $k_f = 1.1$ is shown again in Figure 6-9, 6-10 and 6-11.

In Figure 6-9, a zoomed-in plot of the steering angle is shown. From this figure it is clear to see that the steering output of Sim1 is not a smooth response. This is the result of the same behaviour in the heading error of Figure 6-10. It appears that this problem is not a result of simulation, but that the problem is already present in the path itself. The heading of the path is calculated after the interpolation of the path, this means that the distance between the points is very small. As a result of this small distance the heading error has a little fluctuation.

This problem can be solved in multiple ways:

- One could be to change the path in the heading, this can be done by interpolating the path for a higher distance between points, then calculating the heading, and interpolating the whole path again to get the same smaller distance.
- Another option is to filter the heading error inside the system. This can be done with a Virtual Instrument (VI) that takes a number of last values and computes the average.

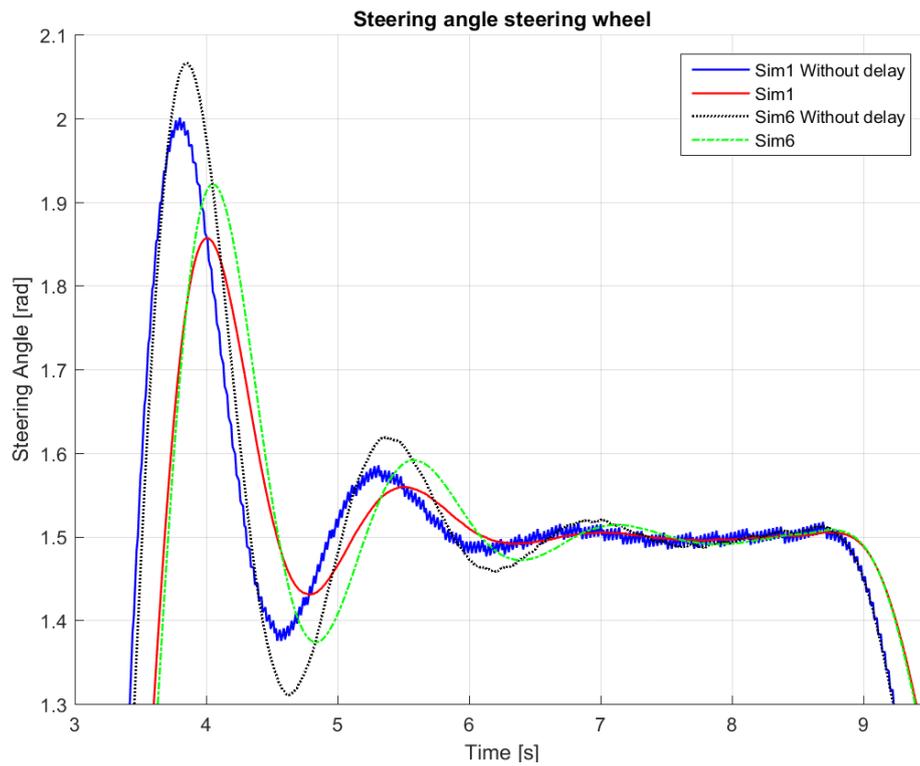
When taken into account that the heading of the GPS is also fluctuating, as was discovered when doing the parameter tests, option two is chosen. This will make sure that the steering angle will not only be smooth in simulation but also in real tests.

- Sim1 is with the values that are previously described, $k_s = 0.7$ and $k_f = 1.1$. These are the parameters that generate the best response.
- Sim6 is with the same values as sim1, but it filtered with the Mean VI for the last 10 values.
 - From the results it can be concluded that the response of the steering is slightly slower when filtered.
 - When looking at the Table 6-4 it is clear that the performance is only slightly deteriorated. The RMS value is the same, and the min and max values of y_e are just a tiny bit larger.

This filter does result in a better steering response with only a small deterioration of performance. Therefore it will be used in the rest of this thesis.

Table 6-4: Evaluation parameters for evaluation of steering output

| | | Simulation 1 | Simulation 6 |
|----------------|---------------------|--------------|--------------|
| RMS | [m] | 0.052 | 0.052 |
| Max y_e | [m] | 0.097 | 0.104 |
| Min y_e | [m] | -0.099 | -0.100 |
| Sum of $ y_e $ | [m] | 55.458 | 55.508 |
| Max a_y | [m/s ²] | 2.282 | 2.357 |

**Figure 6-9:** Steering angle responses for evaluation of steering output

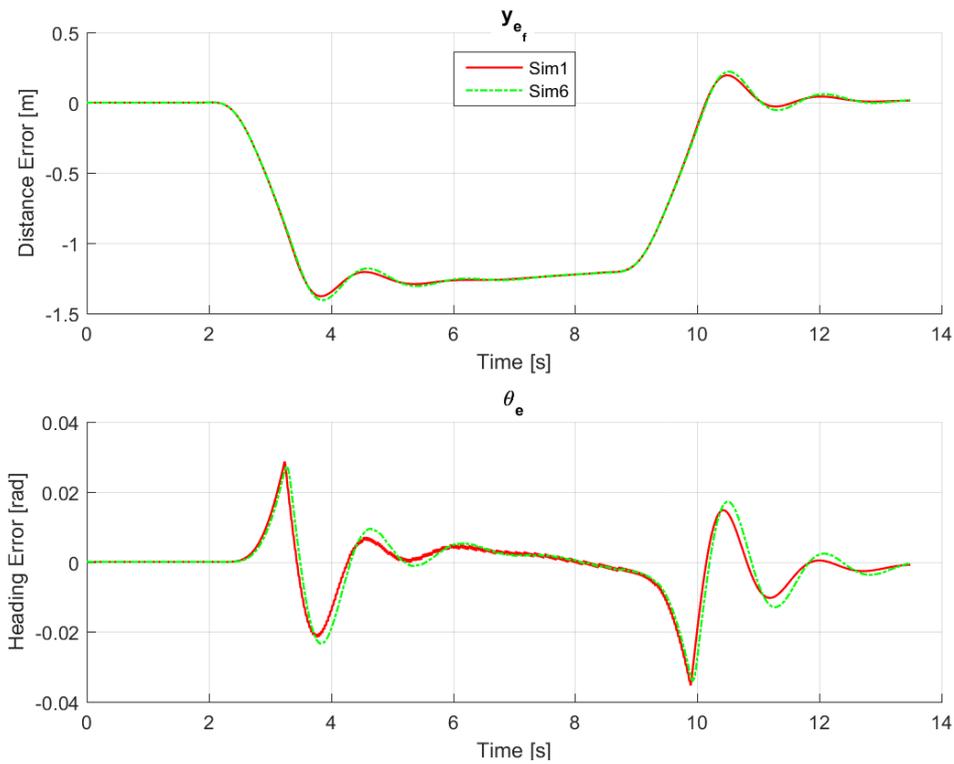


Figure 6-10: Control input for different simulations for evaluation of steering output

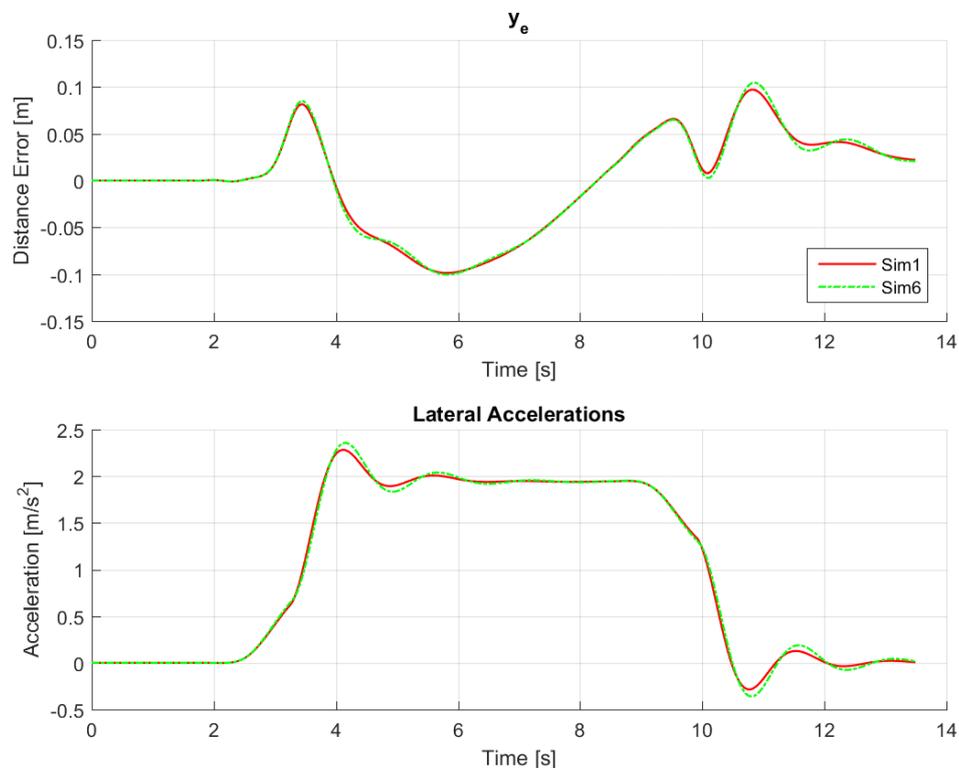


Figure 6-11: Evaluation data of different simulations for evaluation of steering output

6-3 Variable Look-Ahead Distance L_f

An improvement of the control algorithm that was introduced in chapter 4 was the variable look-ahead distance L_f . The idea was that this distance should scale with the forward velocity. To test if this does indeed increase the performance of the controller 2 other simulations are performed.

In figures 6-12, 6-13, 6-14 and 6-15 results are shown from two simulations on a curve with a radius of 55 meters at a speed of 40 kilometres an hour.

- Sim7 is done with the same control parameters as before. $k_s = 0.7$, $k_f = 1.1$ and $k_h = 1$. The length of L_f is taken as the same length as the previous simulations.
- Sim8 has the same control parameters as sim7 but has a feed-forward length that depends on the forward velocity. These simulations are done with a higher velocity and therefore the feed-forward length is larger than previous simulations.

The results are again shown in the figures and can be evaluated with use of Table 6-5.

- From this table it is clear that the variable look-ahead distance is a success. The RMS error is smaller, the sum of error distance y_e is smaller and also the lateral acceleration is smaller.
- The performance isn't as good as the previously simulations. This is a result of the higher velocity. For both simulations it can be observed that it takes longer before a steady steering angle is reached. The larger oscillation of Sim8 can be explained by the larger look-ahead distance. As was shown before in 6-2-2 a L_f that is too large will result in a more fluctuating steering response. This explanation does not hold for Sim7 because that look-ahead distance is exactly the same as before so this should also be influenced by the velocity of the vehicle.
- The steering angle response is lower than the previous simulations. This is a result of a larger corner that is less sharp than before.

The sum of errors is a parameter that can not be used to compare these simulations with simulations that are done before. The trajectories of these simulations are different, and when the total distance is larger the sum of errors will also be larger.

Table 6-5: Evaluation parameters for variation L_f

| | | Simulation 7 | Simulation 8 |
|---------------------------|---------------------|--------------|--------------|
| Feed-forward length L_f | [m] | 9.2 | 12,2 |
| RMS | [m] | 0.393 | 0.285 |
| Max y_e | [m] | 0.209 | 0.559 |
| Min y_e | [m] | -0.797 | -0.366 |
| Sum of $ y_e $ | [m] | 400.703 | 332.365 |
| Max a_y | [m/s ²] | 2.6566 | 2.5432 |

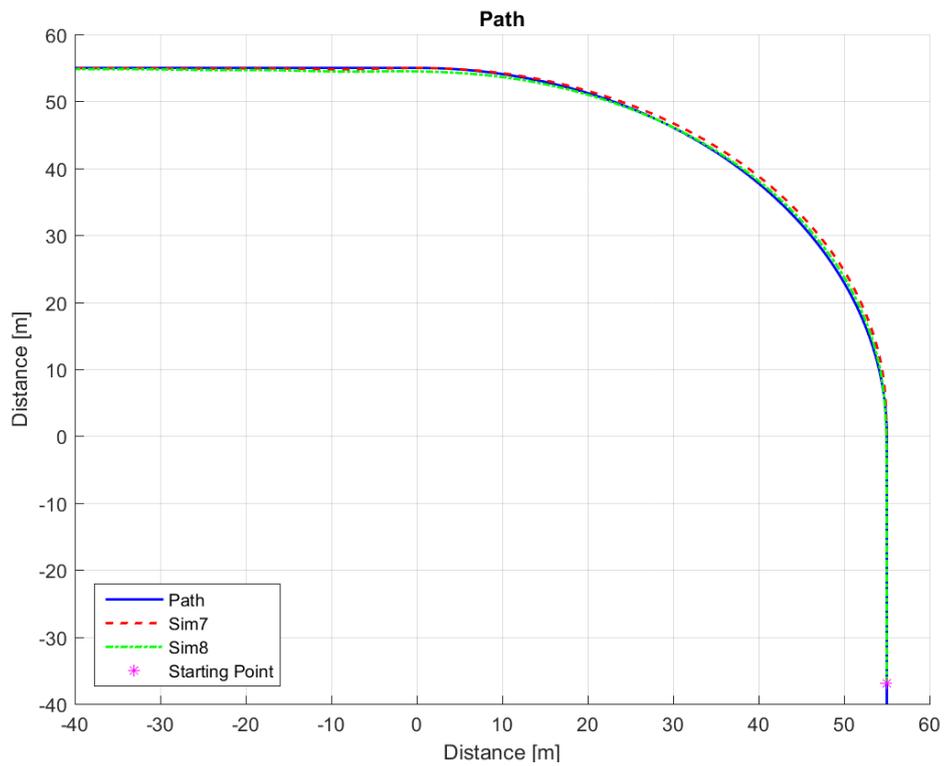


Figure 6-12: The path and simulation responses of variation L_f

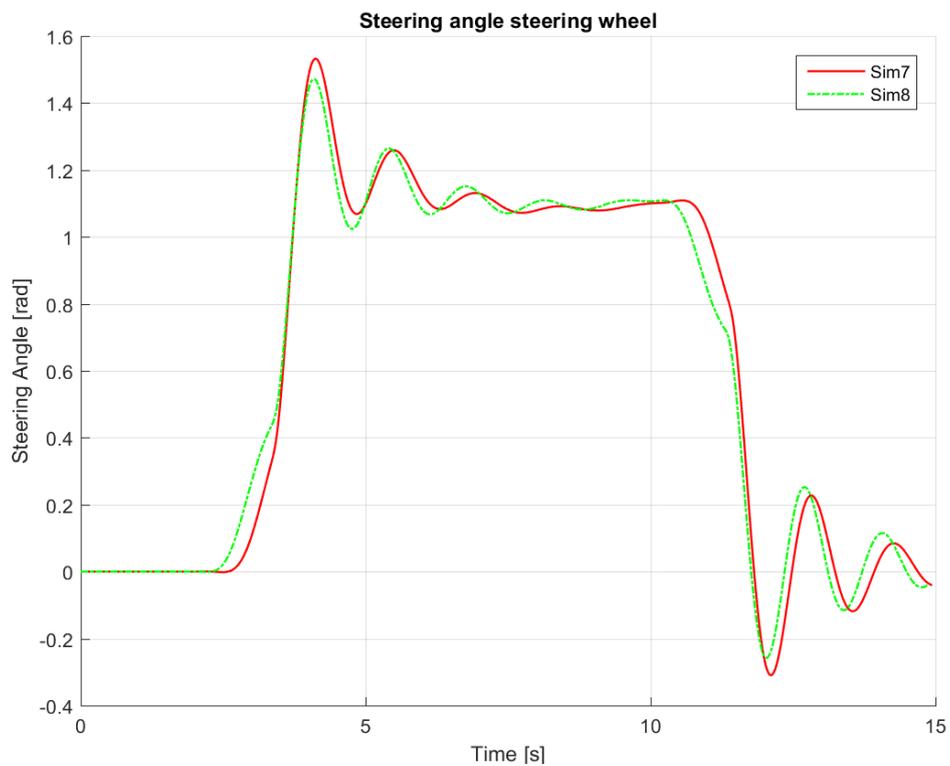


Figure 6-13: Steering angle responses of variation L_f

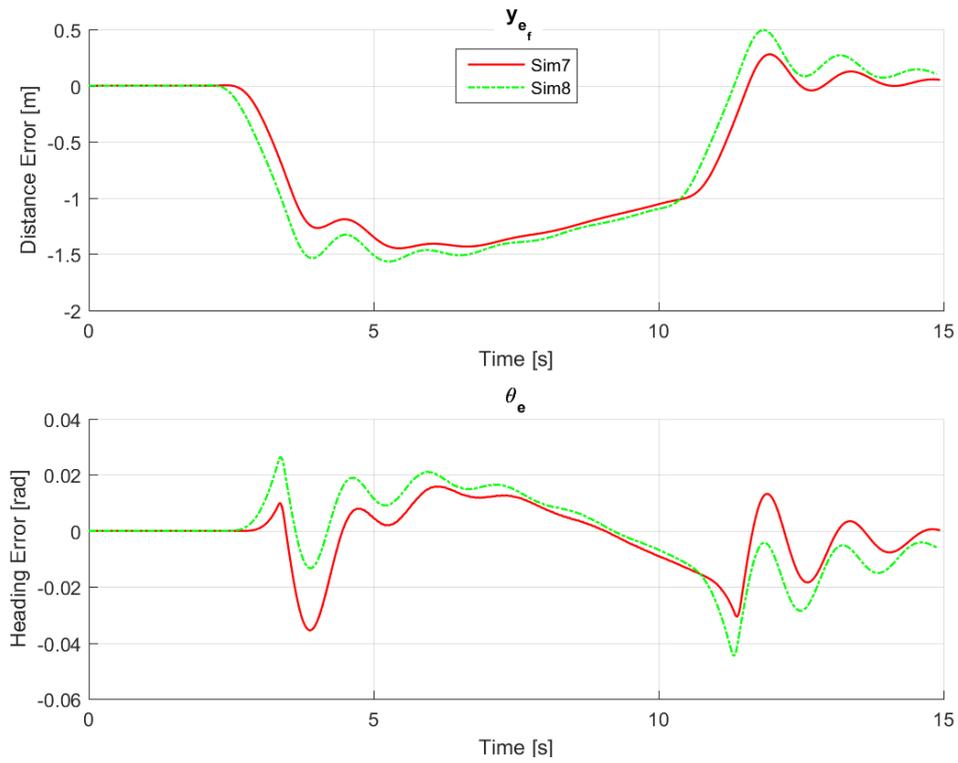


Figure 6-14: Control input for different simulations of variation L_f

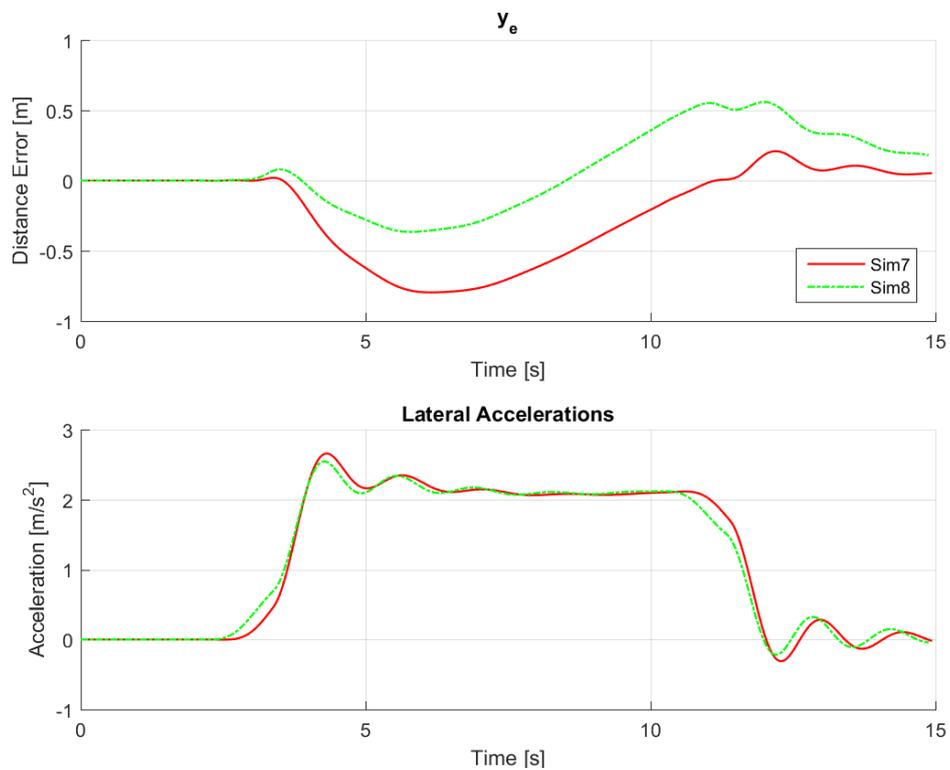


Figure 6-15: Evaluation data of different simulations of variation L_f

6-4 Higher Velocity

The last simulations of this chapter will be simulations with a higher velocity. For this a curve with radius of 85 meters is constructed. The velocity for this trajectory will be 50 kilometres an hour.

- Sim9 is a simulation with the same parameters as before $k_s = 0.7$, $k_f = 1.1$ and $k_h = 1$.
 - The error distance is almost within one meter but the steering angle is oscillating quite a bit. This is the result of an early steering manoeuvre like in section 6-2-2.
 - The positive heading error θ_e starts an oscillation. This means that at higher velocities this controller becomes more unstable.
- Sim10 has the same parameters as Sim9 only the heading parameter is changed to $k_h = 0.4$
 - To solve the instability the heading control parameter is lowered.
 - From the figures and the table it can be seen that this does improve the performance of the controller.
 - The steering angle response is much more stable and the other evaluation parameters are better than sim9.
- Sim11 has different parameters than the other simulations $k_s = 1.1$, $k_f = 0.8$ and $k_h = 0.4$.
 - With these different control parameters the response of the controller is better.
 - From Table 6-6 it can be seen that the minimum and maximum distance error becomes smaller. Also the RMS value is lower.

As a result it can be concluded that the steering response is highly depended on the velocity of the vehicle, and the thereby corresponding corner.

Table 6-6: Evaluation parameters for higher velocity

| | | Simulation 9 | Simulation 10 | Simulation 11 |
|----------------------|-----------|--------------|---------------|---------------|
| Steering Parameter | k_s | 0.7 | 0.7 | 1.1 |
| Look-Ahead Parameter | k_f | 1.1 | 1.1 | 0.8 |
| Heading Parameter | k_h | 1.0 | 0.4 | 0.4 |
| RMS | $[m]$ | 0.537 | 0.429 | 0.331 |
| Max y_e | $[m]$ | 1.012 | 0.898 | 0.549 |
| Min y_e | $[m]$ | -0.725 | -0.569 | -0.640 |
| Sum of $ y_e $ | $[m]$ | 720.806 | 559.203 | 426.850 |
| Max a_y | $[m/s^2]$ | 2,474 | 2.402 | 2.545 |

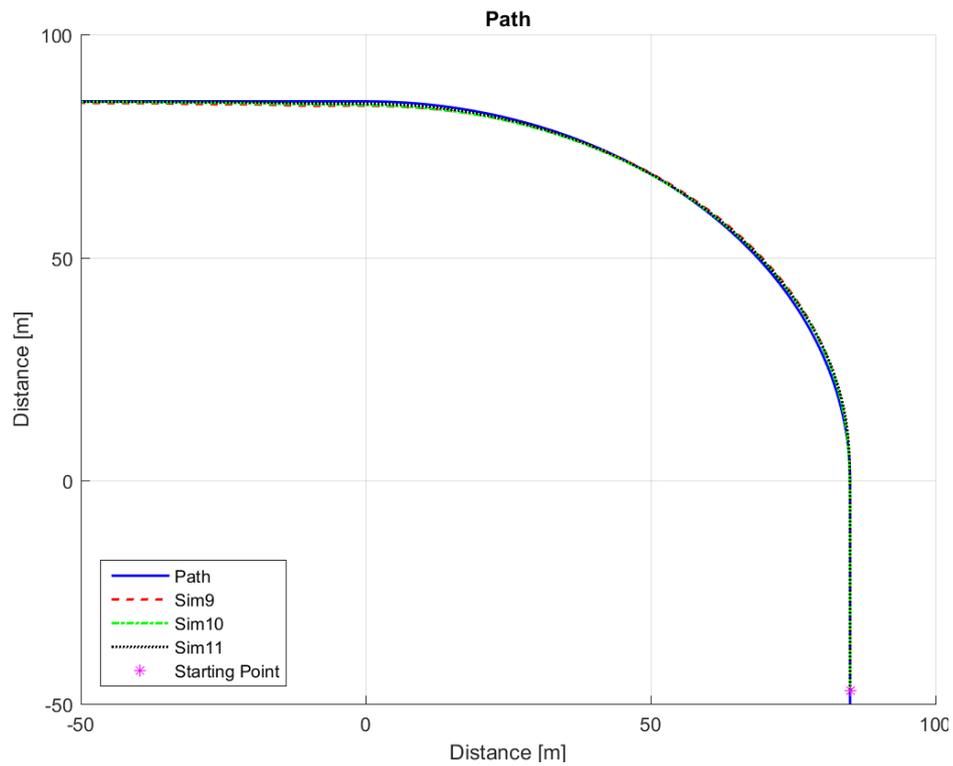


Figure 6-16: The path and simulation for higher velocity

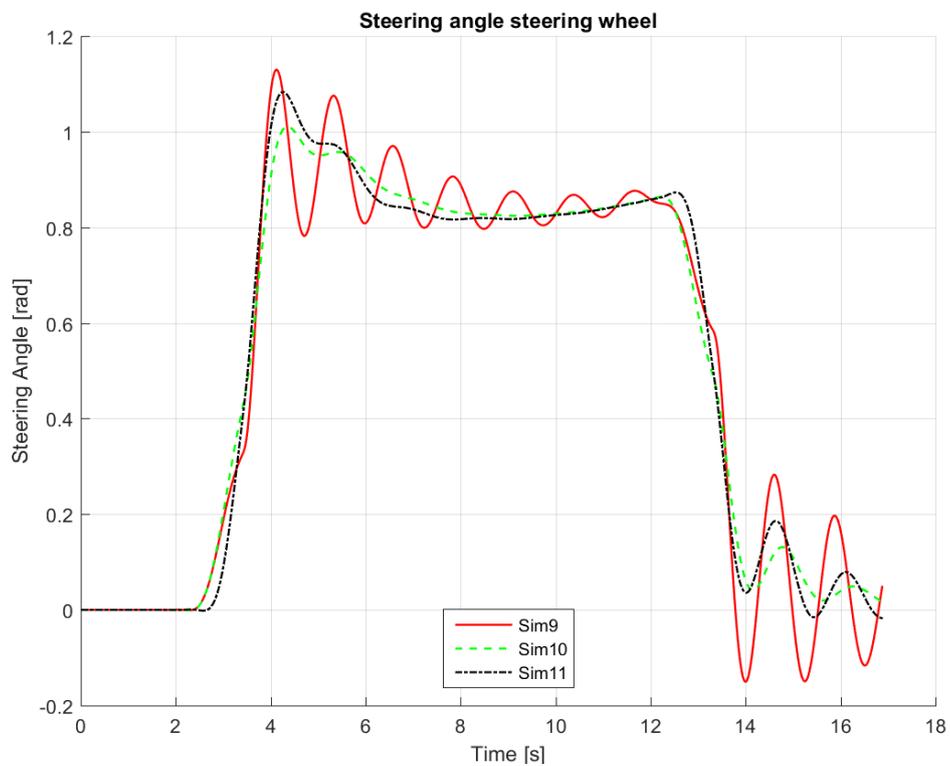


Figure 6-17: Steering angle responses for higher velocity

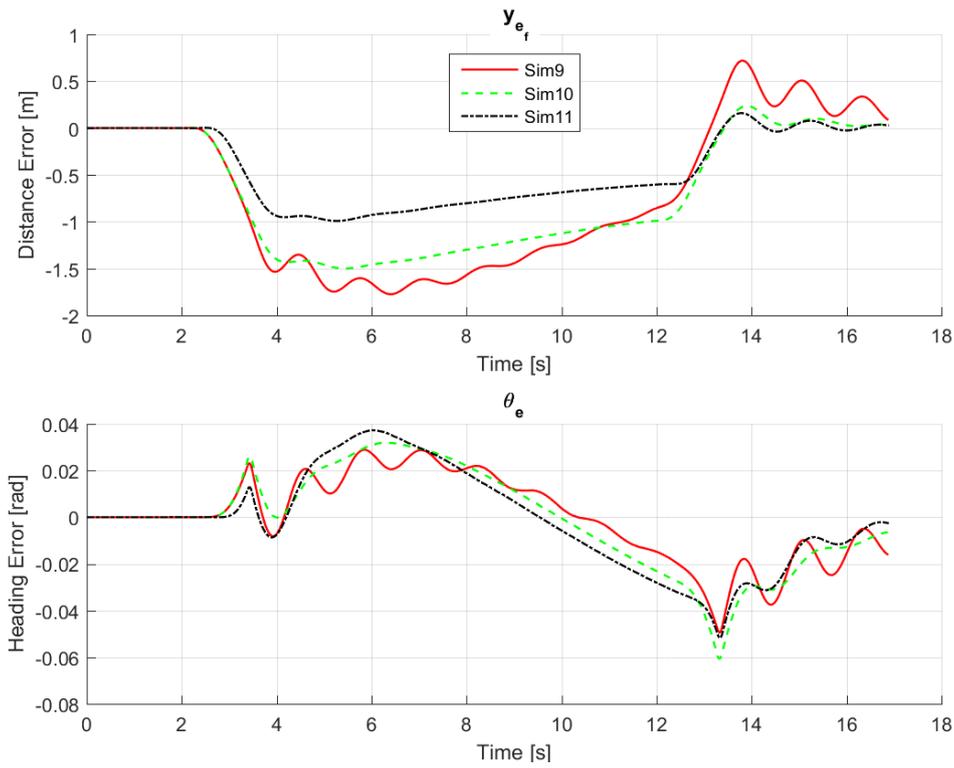


Figure 6-18: Control input for different simulations for higher velocity

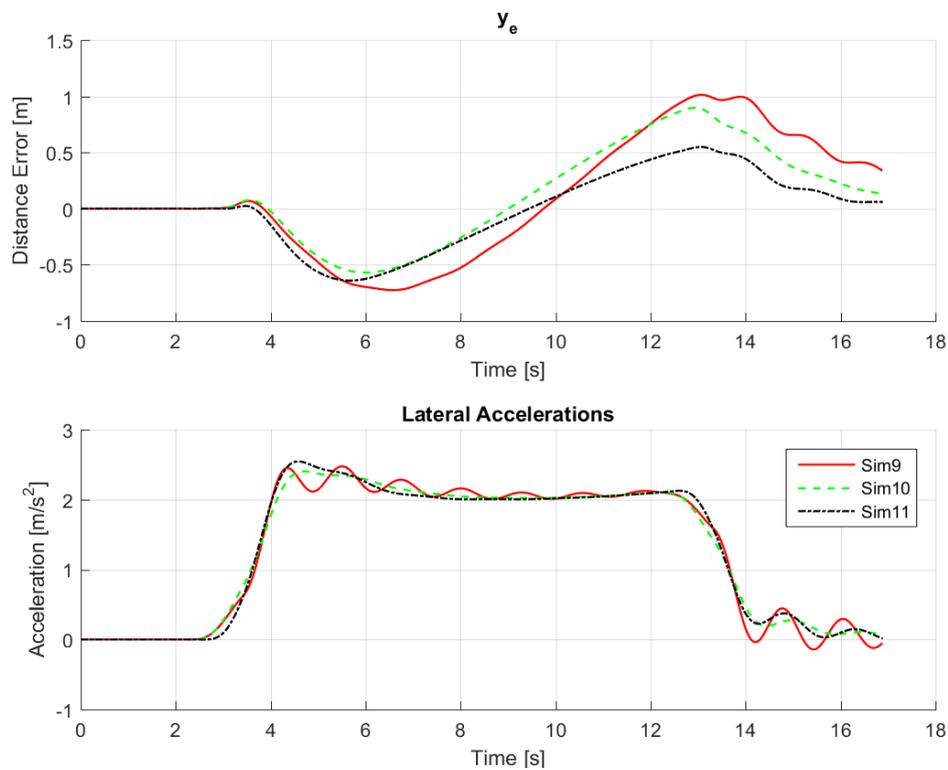


Figure 6-19: Evaluation data of different simulations for higher velocity

6-5 Discussion

In this chapter simulations of the path follow system are performed. The simulations were done at a frequency of 100 Hz to simulate the system at a high frequency that is the same as vehicle safety systems like ABS or ESP. The performance of the controller was evaluated with a variety of constant velocities at different curves. In the first section of this chapter it is shown that the path follower performs very good at a velocity of 30 km/h. In simulation the vehicle stayed within 20cm of the path.

After the first simulations it was observed that the heading angle was fluctuating because of a little disturbance in the vehicles trajectory. Bearing in mind that the actual heading of the Global Positioning System (GPS) would also have some disturbance, the choice was made to filter the heading error with a mean function. This resulted in a much more stable steering wheel response at the cost of a very small performance setback. As was shown in section 6-2-3.

Next to that the improvements of the control algorithm of literature, that were introduced in chapter 4, were validated. The variable look-ahead distance does indeed help the controller to perform better when faced different forward velocities. Although this only holds for a limit velocity variation. If the change in velocity becomes to big only changing the look-ahead distance doesn't result in the perfect tracking. A solution for this would be to have a set of control parameters for groups of velocities. For instance a control parameter from 10 to 20 km/h, one for 20 to 30 and so on. These groups could be bigger or smaller depending on the change in performance. This could be a good research topic on its own and this idea would not be further researched in this Thesis.

For higher velocities it was concluded that the steering response was too much influenced by the heading error. Therefore the improvement introduced in chapter 4 was also working. It was proposed to give the heading error a control parameter, and as showed in section 6-4 it did result in a much more constant steering angle output of the path follower. It is possible that this phenomenon does become larger at higher velocities but that will not be investigated in this thesis.

The simulations of the path planner are finished and discussed. The next step is to evaluate the performance of the path follower on the real vehicle. This will be done in the next chapter. Also the difference between running the system at the safety frequency of 100 Hz and the vehicle's frequency of 25 Hz will be discussed.

Chapter 7

Real Tests

This chapter will compare the results of simulation with results of tests on the real vehicle. In the beginning of this chapter the different process frequency is discussed. Then the trajectory of the tests is introduced. The test results are shown and compared. The creation of the trajectories heading is also evaluated. In the end of this chapter some other tests results are shown with a higher velocity. Finally the chapter will end with a discussion about the results.

7-1 Frequency

The system of the vehicle system has a different frequency than the simulations that were done before. The maximum frequency of the vehicles parameters like the velocity and position is $25Hz$ as was discussed in chapter 1. Due to some unexplained fault in the system the test data that is used in this thesis was saved at a frequency of $12.5Hz$. This is quite lower than the simulations that were performed earlier. The difference between these frequencies can be shown with use of simulation.

In figures 7-1, 7-2 and 7-3 the results of two simulations are shown.

- Sim6 is the same simulation as the previous chapter with the same parameters as before, including the heading filter and at $100Hz$.
- Sim12 is done with the same parameters but at a lower frequency, the frequency is $12.5Hz$.
 - The lower frequency is realised by holding the control outputs with a zero order hold function. In this way there is no important information lost for the simulation of the vehicle but the controller will work on a lower frequency.
 - The zero order hold is visible in the response of the steering wheel. When the signal is observed closely it is visible that the steering wheel angle stays constant for a small period of time.

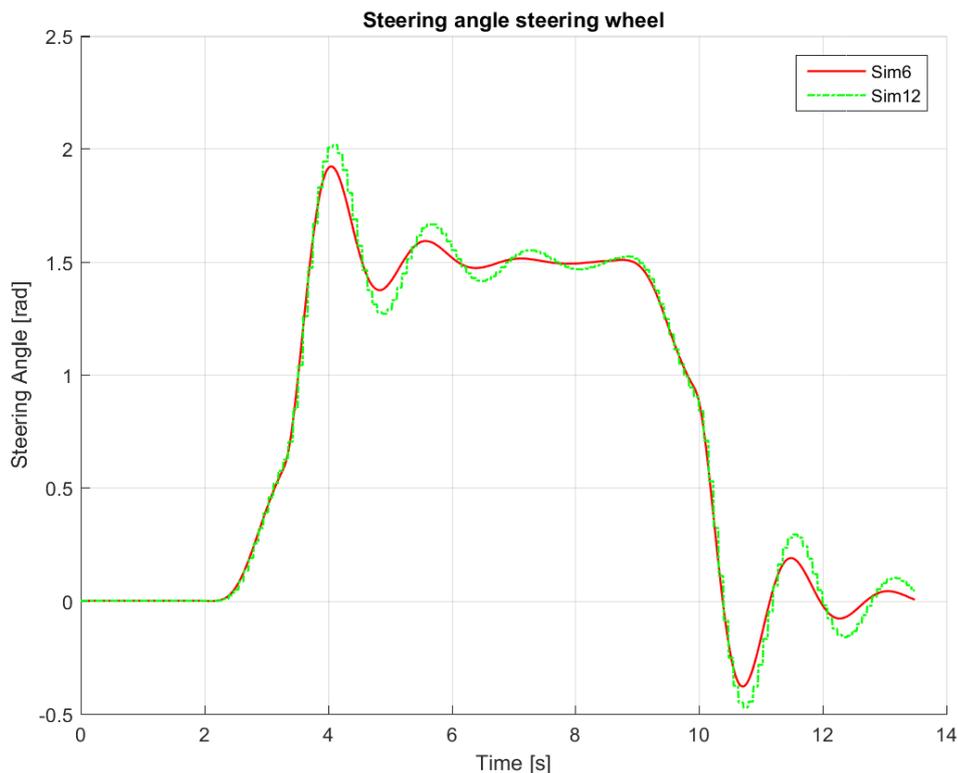
Table 7-1: Evaluation parameters for evaluation of different frequencies

| | | Simulation 6 | Simulation 12 |
|----------------|---------------------|--------------|---------------|
| Frequency | | 100Hz | 12.5Hz |
| RMS | [m] | 0.052 | 0.058 |
| Max y_e | [m] | 0.0104 | 0.129 |
| Min y_e | [m] | -0.100 | -0.112 |
| Sum of $ y_e $ | [m] | 55.508 | 60.569 |
| Max a_y | [m/s ²] | 2.357 | 2.487 |

The differences between the two simulations are shown below.

- When looking at the performance values shown in Table 7-1 it can be noticed that the simulation with a lower frequency has a larger error.
- With lower frequency the Root Mean Square (RMS) Value is slightly larger and the maximum and minimum distance errors are larger.
- The larger error can entirely be explained by the slower response of the controller.

Even though it is clear that the response of the system becomes a little bit worse it isn't that bad at all considering the fact that the controller takes 8 times as much time to react to changes in the trajectory.

**Figure 7-1:** Steering angle responses for evaluation of different frequencies

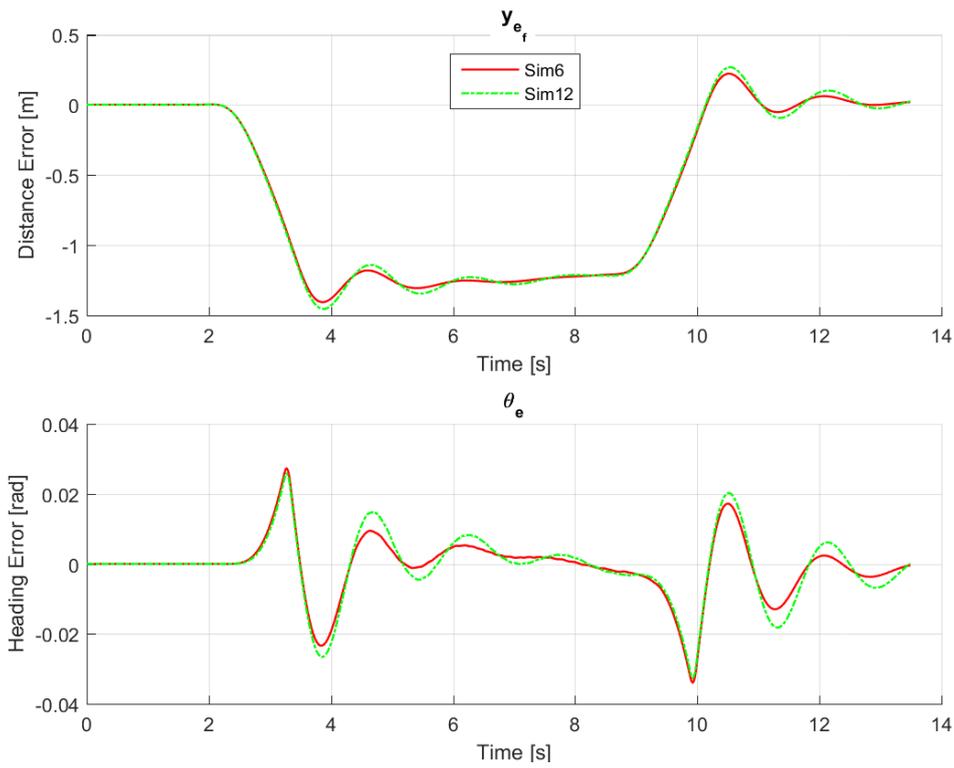


Figure 7-2: Control input for evaluation of different frequencies

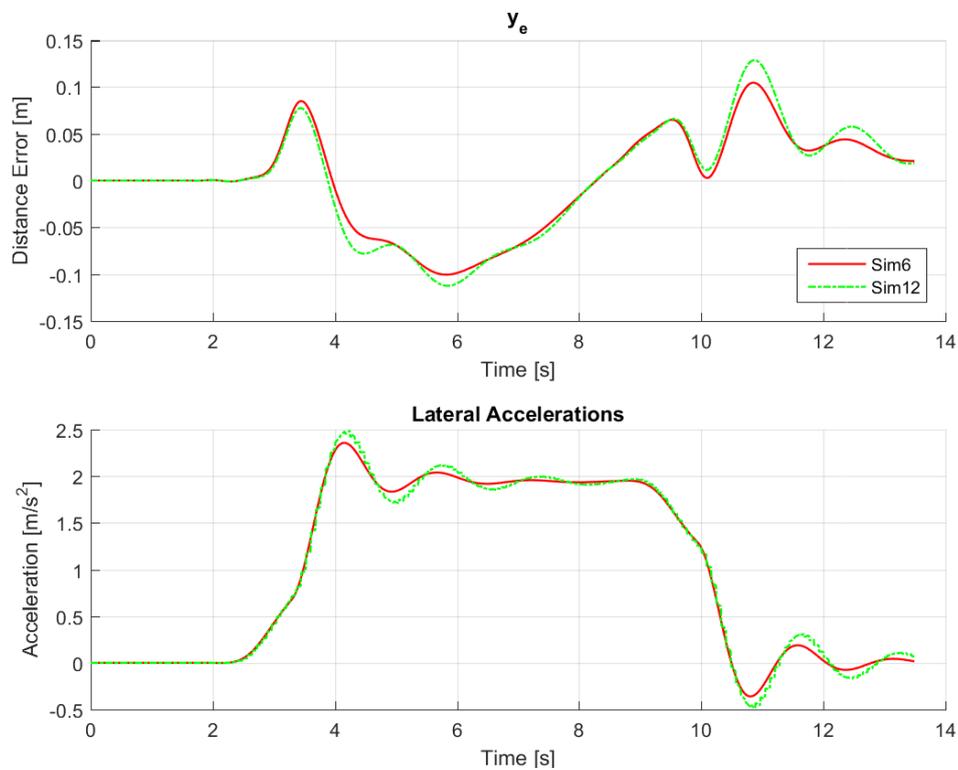


Figure 7-3: Evaluation data for different frequencies

7-2 Trajectory

Before it is possible to test the vehicle it is necessary to have a valid trajectory. As discussed before there are multiple ways to generate a trajectory.

- The first option is to design a path in Matlab.
Before it is possible to put these paths on the real system it is necessary to change the position to a valid Global Positioning System (GPS) location.
- The other option is to drive a path with the actual vehicle and save the GPS data while driving the trajectory.
This will create a list of GPS positions and a heading that can be used as input for the path follower. Before it can be used it needs to be interpolated to have a fixed distance between every point and also some inconsistencies need to be removed.

This last option is used for the trajectory that will be used in the rest of this chapter. This path was driven at Valkenburg Naval Air Base, which is a closed area that could be used to safely test the path follower. The trajectory was driven at low speed to generate a good number of data points for the path.

The path is shown in Figure 7-4 on the next page, the next points hold for path:

- Because this path is driven with the actual vehicle the end and beginning of the path do not match exactly. This results in a small gap between the two lines.
- The corners of this path are not exactly the same. This may result in some difference error responses in different corners. But it is also a good test to show how the controller performs at different corners.
- The lines between the corners aren't completely straight but they have a small variation.

The velocity of the path will also be variable. After preparing the path for use a velocity profile is added. This is done by selecting the maximum acceleration values, which are chosen from the literature described in chapter 2. The acceleration value will be $1.5m/s^2$ and the deceleration $-2m/s^2$. Then suitable velocities were chosen, for this path a maximum velocity of $15km/h$ and a cornering velocity of $10km/h$. The profile is visible in Figure 7-5.

The only thing left to do is to determine the change of velocity. This was done by calculating the distance necessary to accelerate or decelerate with the previously described values. For instance accelerating from 10 to 15 kilometres an hour would take:

$$\begin{aligned}\Delta V &= A * t \\ 1,3889 &= 1.5 * t \quad t = 0,93[s]\end{aligned}\tag{7-1}$$

And with that time it is possible to calculate the distance:

$$\begin{aligned}D &= v_1 * t + A * t^2 \\ D &= 2.7778 * 0,93 + 1.5 * (0,93^2) \quad D = 3.22[m]\end{aligned}\tag{7-2}$$

After calculating the required distances, points were selected on the path to start and stop these accelerations. The points on the path were selected in a way that the vehicle would brake before the turn and start to accelerate after the turn. These points are visible in Figure 7-4.

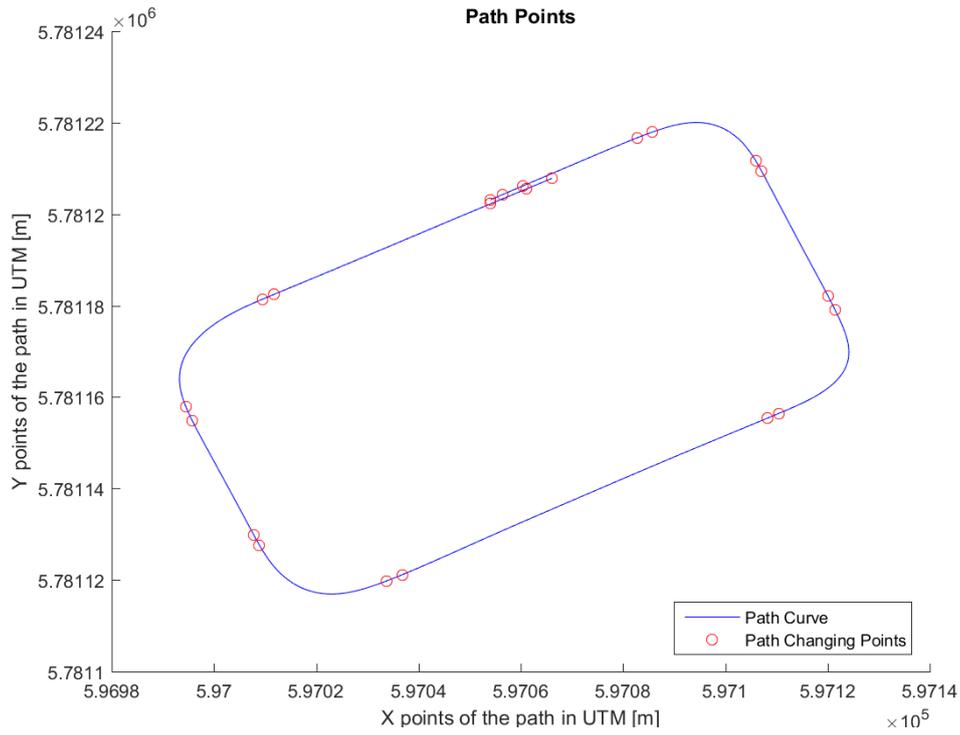


Figure 7-4: Trajectory for the test drive

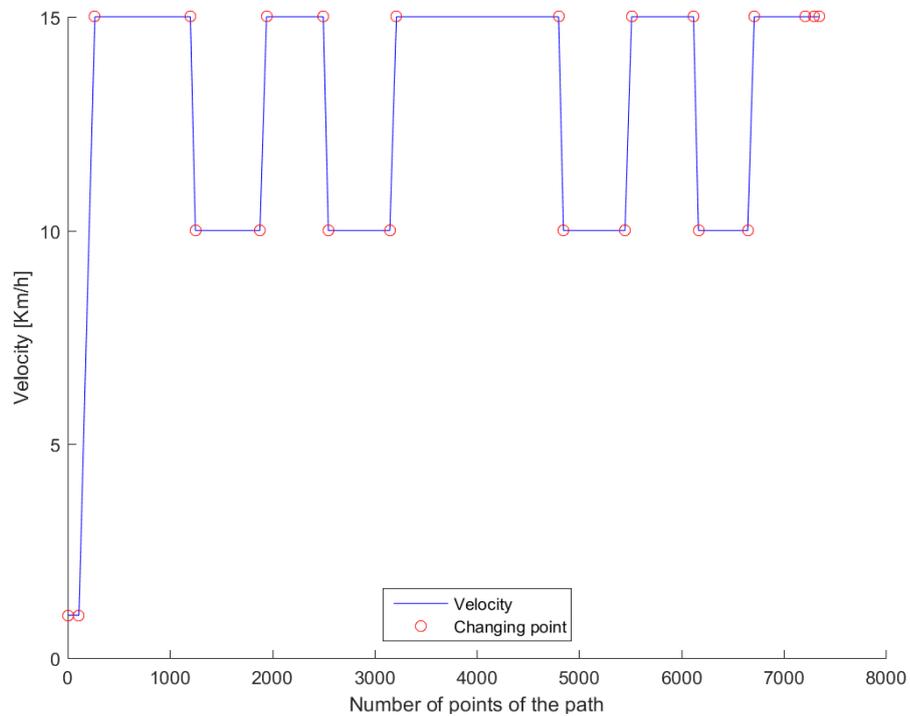


Figure 7-5: Velocity profile for the test drive

7-3 Heading

Before the simulation results are compared with the vehicle test there is one thing left to do. This is deciding which option for the heading is best. In chapter 5 different options for the design of the heading parameter were discussed.

- The first heading option is to measure it with GPS and then interpolate this to the full length of the path.
- The other option is to calculate the heading from the interpolated path.

For the previous simulations it was not possible to choose the measured value because these simulations were performed on a path that was constructed in Matlab. The path that is described in the previous section is driven with the vehicle and therefore a GPS heading is available.

The first tests that were performed with the vehicle were to decide which option would result in a better response. In the figures 7-6, 7-7, 7-8 and 7-9 the results from 2 tests are shown.

- The first test is with a calculated heading.
- The second test is with a heading that was logged from the vehicle.
- Both tests were done with the same control parameters.

The difference between the two tests is quite clear.

- From Figure 7-6 it can be observed that the second test cuts the final two corners of the trajectory.
- From Figure 7-7 it is visible that the steering response of the second test is much more unstable than the steering response of the first one. This can also be observed at the heading error θ_e and distance error y_{ef} where there is an oscillation.
- From Figure 7-8 it is clear to see that the two control values are giving a different sign value. One wants the vehicle to steer left and the other one wants to steer the vehicle to the right.

The result of these tests is that it can be concluded that the heading that is calculated from the trajectory will have a much better response than with the logged heading. This result difference could be a result of lack of satellite connections at the time of logging the path. But this would also have resulted in a more shattered line of path points. It is also possible that there was some other fault in the system, for instance the antennas weren't placed correctly and as a result the system calculated the wrong heading.

As a conclusion of this section it can be noted that, even though the reason for this difference in performance is unknown, it is clear that when designing a trajectory it is better to calculate the heading from the trajectory than to use a logged one.

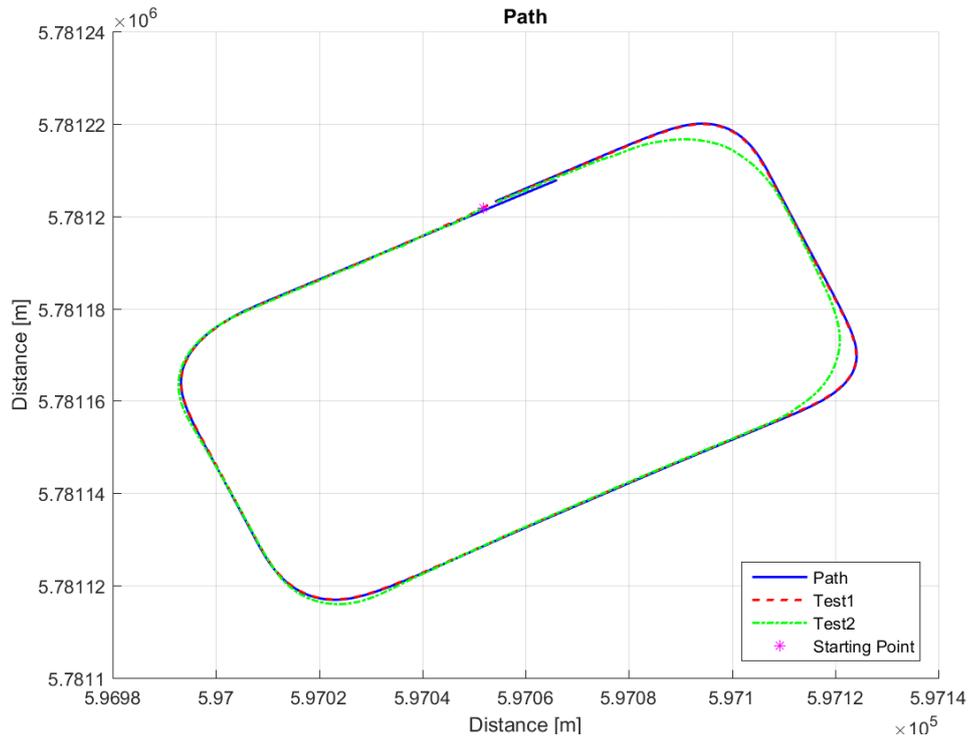


Figure 7-6: The path and test responses for different headings

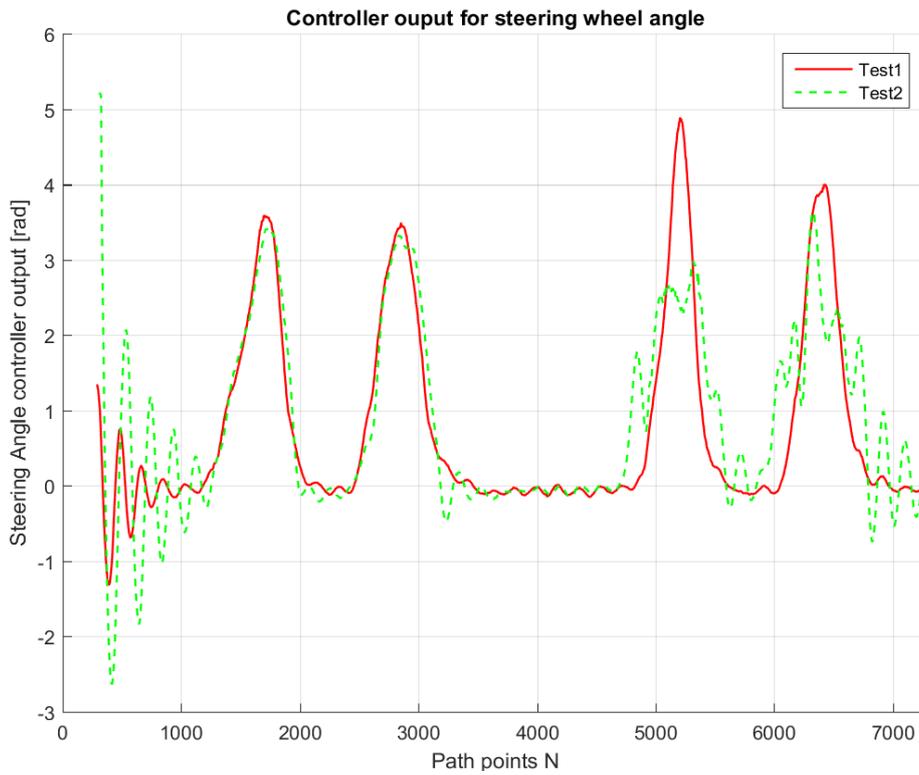


Figure 7-7: Steering angle responses for different headings

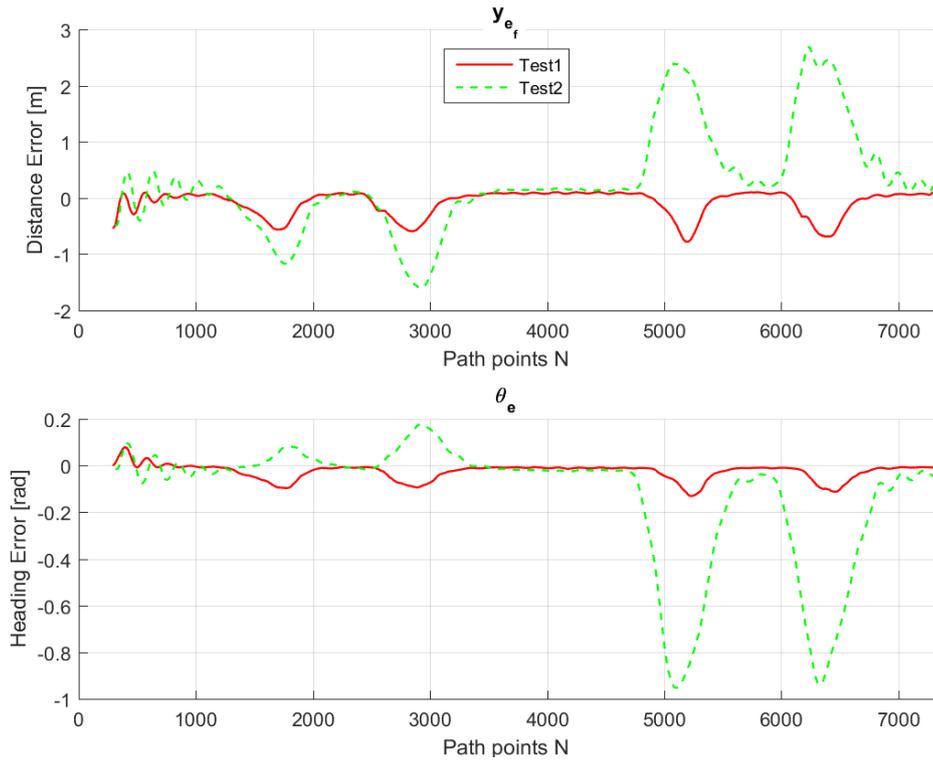


Figure 7-8: Control input for different headings

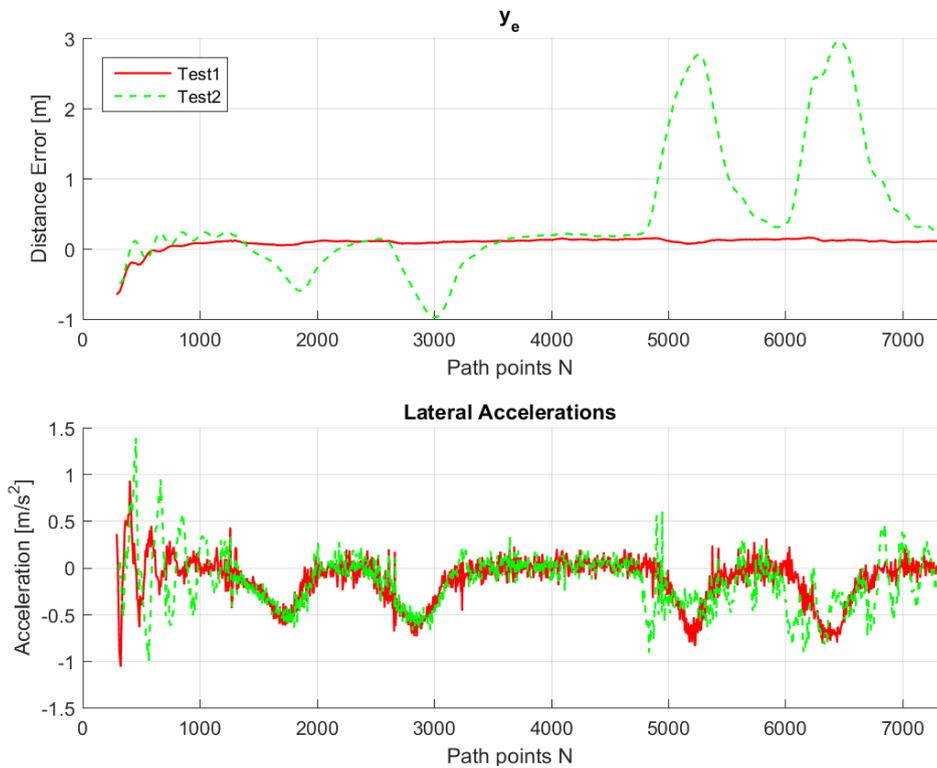


Figure 7-9: Evaluation data of different headings

7-4 Test Drive

The previously described trajectory is used in this section for a test with the Toyota Prius. The results of this test can be seen in the figures: 7-10, 7-11, 7-12, 7-13 and 7-14. Instead of being plotted on the time the figures are now plotted on the path point number. This is a result of different velocities that would make it impossible to plot the results in the same graph and still make it possible to say something about them.

- Simulation 100Hz is a simulation at a frequency of 100Hz.
- Simulation 12.5Hz is a simulation at a frequency of 12.5Hz.
- Test 12.5Hz is a real test with a frequency of 12.5Hz.
- The previously optimized control parameters were used to see the control results on a different path. $k_s = 0.7$, $k_f = 1.1$ and $k_h = 1$.

The following observations can be made from the figures:

- There is some fluctuation in the beginning of the responses.
This is due to the fact that it was hard to find the perfect starting point of the path. Therefore it was chosen to start somewhere on the path and let the vehicle drive the whole trajectory. When it had reached the end it would automatically start over, but there is a gap between the end and beginning of approximately 50cm. Therefore the simulations and also the real response starts with an error.
- The the real test results are much better.
When looking at Table 7-2 it can be observed that the error values are smaller for Sim3. The total error parameter can not be compared, because the real test data exists out of fewer points. Therefore the sum of there error values will always be smaller.
- The error distance y_e stays much smaller in the real test.
Also it is much more consistent. It is roughly a straight line, when not considering the first part.
- The difference between the simulation and real test is best shown in Figure 7-12, where the heading error is shown.
The difference is clear at the second and last corner of the trajectory. In the simulations the heading error will become positive, which shows that the vehicle is overturned. This is a result of the larger steering angle. This larger steering angle is required by the control system because the error distance y_{ef} becomes much larger than in the test. In turn this error distance is larger because the vehicle is further away from the trajectory.
- From Figure 7-11 it can be observed that the vehicle and simulation start steering at almost the same path point.
This means that they start steering at roughly the same point in the trajectory, but somehow the result of this steering action is different. This results in a different maximum steering angle, as is shown in the figure. For corners one and three the steering of the simulation is lower, as is the opposite for turns two and four.

- When looking at Figure 7-14 it can be concluded that this difference is not a result of difference in velocity.
The simulation value and vehicle value for velocity are quite similar, especially the lines of acceleration and deceleration which are almost exactly the same. As are the longitudinal acceleration values.
- The simulation and vehicle start steering at the same time and have roughly the same velocity, this means that the difference between simulation and vehicle is a result of some modelled vehicle dynamics. It looks like the real vehicle is turning better than the simulation. This can for instance be a result of weight transfer that is not considered in the model.

Table 7-2: Evaluation parameters for difference between simulation and real test

| Test | | Simulation 100Hz | Simulation 12.5Hz | Real Test 12.5Hz |
|----------------|---------------------|------------------|-------------------|------------------|
| RMS | [m] | 0.332 | 0.333 | 0.122 |
| Max y_e | [m] | 0.113 | 0.114 | 0.158 |
| Min y_e | [m] | -1.193 | -1.195 | -0.654 |
| Sum of $ y_e $ | [m] | 1764.219 | 1780.331 | 131.159 |
| Max a_y | [m/s ²] | 0.981 | 1.014 | 1.063 |

7-4-1 Accuracy of GPS

The accuracy of the GPS is also a point that needs to be discussed. The GPS system includes a standard deviation for latitude and longitude position in meters. These values were available after testing and these values are:

In latitude approximately 0.13 meter and in longitude approximately 0.10 meter.

On average this means that the standard deviation of the GPS is 0.115 meter in the test. This is almost the same as the distance error y_e . This could mean that the vehicle was even closer to the recorded path than shown, or when the standard deviation is added it would be further away. Nevertheless a distance error that is within the standard deviation of the GPS is a good performance of the path follower.

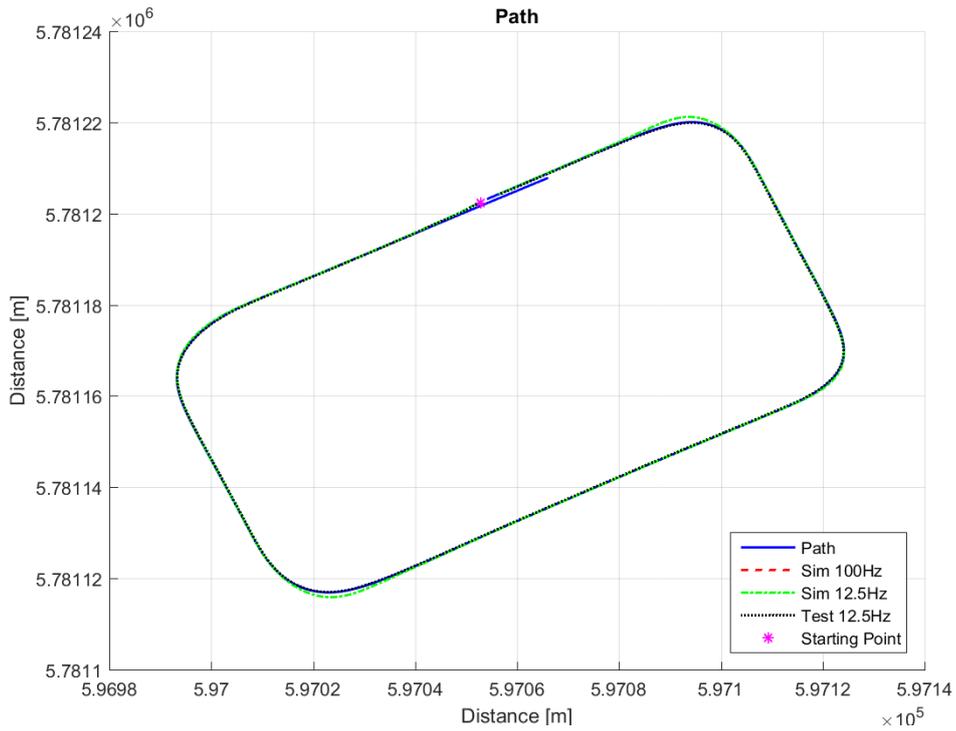


Figure 7-10: The path, simulation responses and the response from the test drive

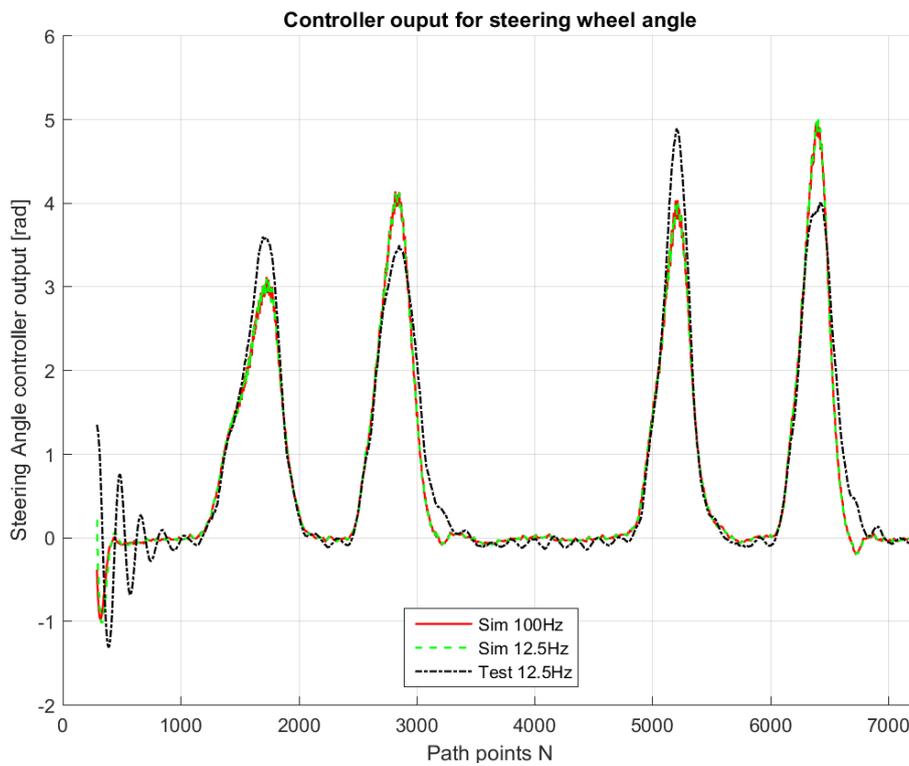


Figure 7-11: Controller output for simulations and test drive

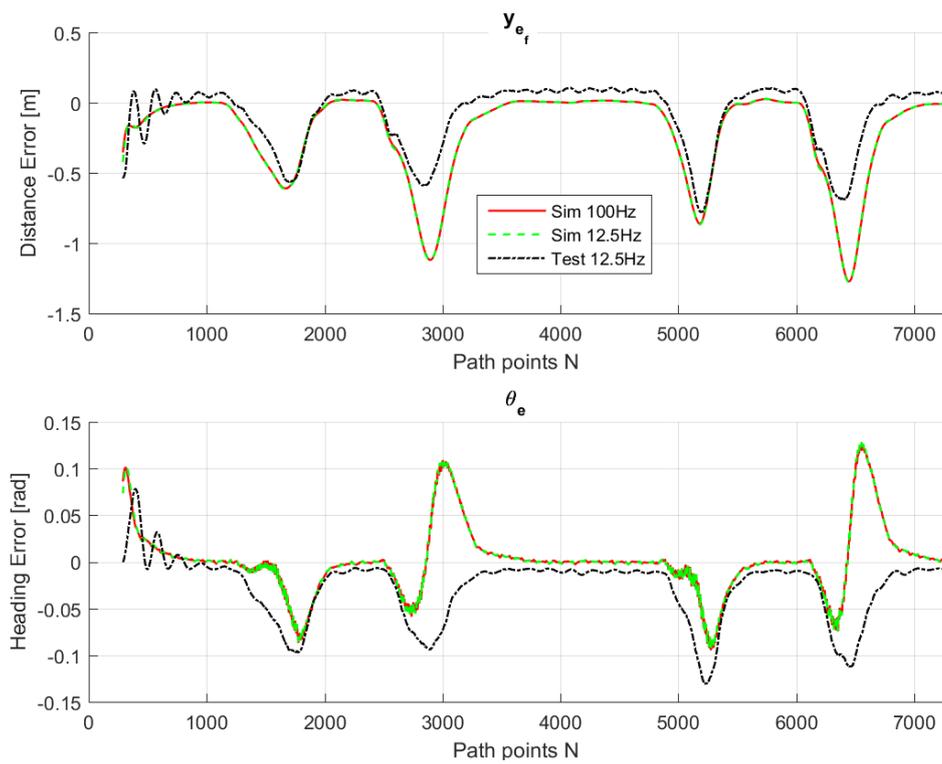


Figure 7-12: Lateral control inputs for simulations and test drive

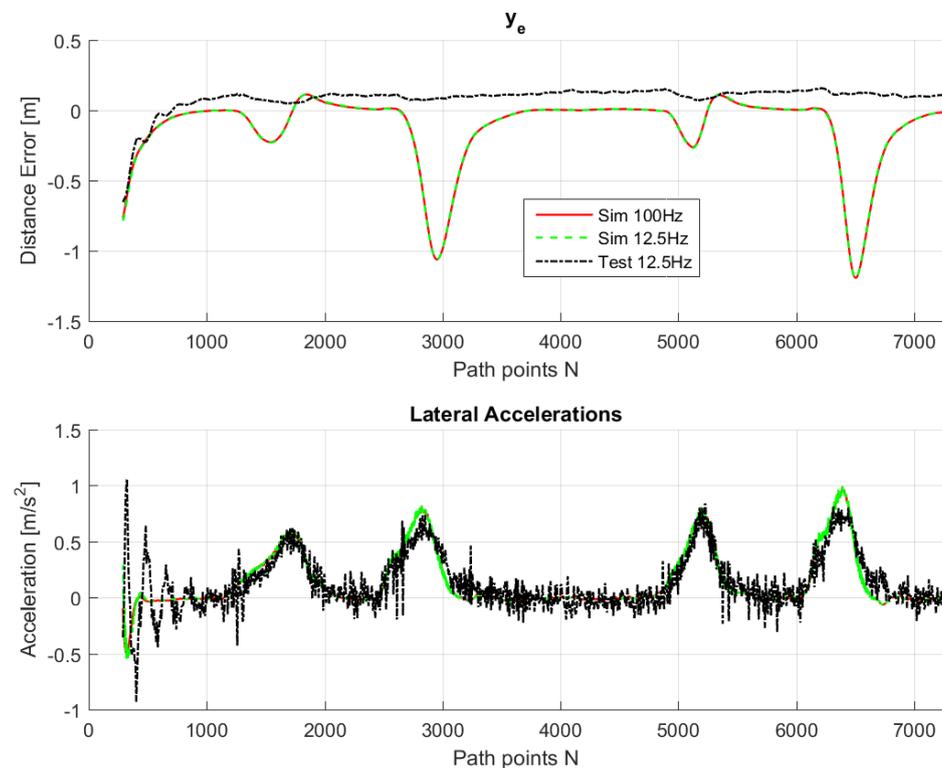


Figure 7-13: Evaluation data of simulations and test drive

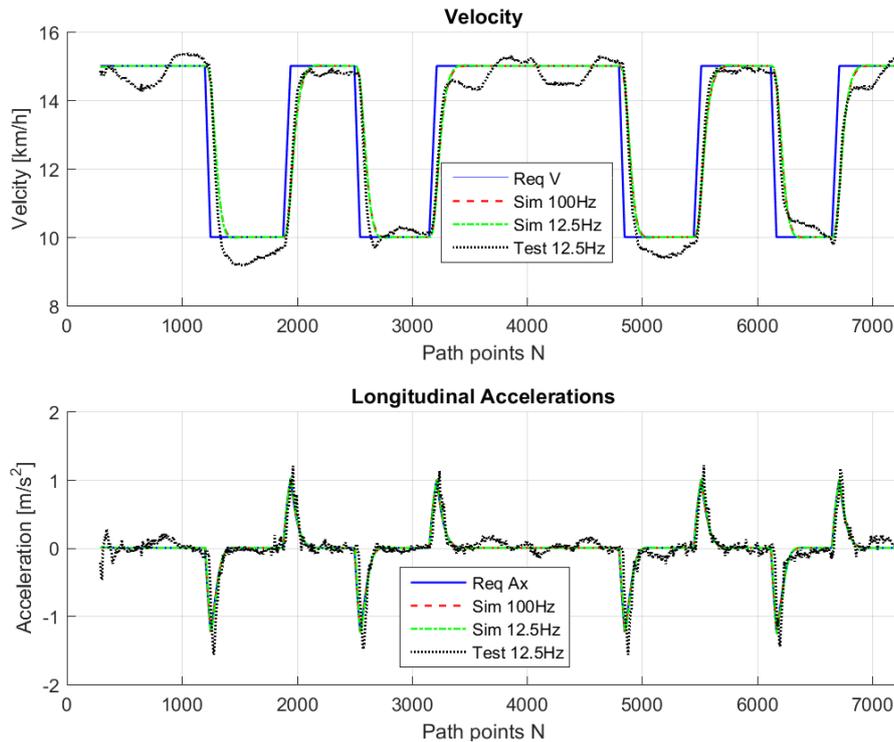


Figure 7-14: Longitudinal control output and resulting velocity for simulations and test drive

7-5 Discussion

This chapter started with the difference between simulation with a frequency of 100 Hertz and a frequency of 12.5 Hertz. From the simulations it could be observed that there is a difference in performance but that this difference wasn't very large. The steering wheel output was a little behind the output of the high frequency. This leads to a bit larger error distance y_e . But this is one simulation at a relative low velocity. It could mean that, because there is a delay in the steering, the performance at higher velocities may deteriorate further at lower frequency. It is also important to note that the vehicle should be able to perform the control at a frequency of 25 Hertz, this means that it is possible to quickly improve the response.

7-5-1 Path

After the difference between frequency was presented and evaluated the trajectory of the test was introduced. As discussed before this trajectory was created with use of logged GPS data. This resulted in a path that doesn't have perfect straight lines or corners. This means that there is a disturbance present in the path itself. Even though it was filtered and interpolated. This means that the performance of the path planner could, in theory, be better when there is no disturbance in the path. This will result in a more stable and continues steering response but probably not in a better tracking because the disturbance is very small.

In the same section a velocity profile was introduced and explained. The velocity change points are positioned in such a way that the vehicle would brake before it starts to steer for

the corner. After the corner is completed the vehicle would accelerate again. This acceleration and deceleration is done with limited values that are found by the use of literature.

7-5-2 Heading

The design of a path is discussed in chapter 5. In this chapter two different options for the heading are described. These two options were evaluated in this chapter. It was clear that the heading that is calculated from the path resulted in a much better response than the interpolated heading.

As discussed before this difference could be a result of an unknown error during the logging process. For instance the misplacement of the antennas. Even though it is not possible to find the source of this error it can be determined that the performance of the interpolated heading is worse. Not only in the last two corners, were the response has a large deviation, but also in the rest of the simulation. Which is best visible in the distance error that is more fluctuating and also has larger error values.

7-5-3 Performance

The final part of this chapter was the comparison between the simulation and vehicle test. The results, which are already discussed, are remarkably better in the vehicle test than in the simulations. Where the simulations have an error of over a meter the real vehicle stays within 16 centimetres of the path. In the evaluation of the performance it was already mentioned that this is a result of the difference between vehicle model and actual vehicle. This difference can be explained by the fact that the vehicle parameters are optimized on a self driven test set. A explanation for the difference could be that this data set is not entirely representative for the actual vehicle. Another explanation could be that the vehicle model that was used did not include all the vehicle dynamics. This could be the reason for the difference, as the vehicle model that was used has a lot of assumptions.

The results of the real test are as previously mentioned almost within the standard deviation of the GPS. This is a very good performance of the path follower, which is much better than the performance that was described in the literature. One of the reasons for this could be that the GPS that was used in this test is an more advanced and more expensive version of the GPS that was used in literature. But it cannot be neglected that the tuning and improvements of the algorithm did also improve the performance.

The longitudinal performance of the path follower is also quite good for a very simple control algorithm. From the figures it can be observed that the longitudinal acceleration values of the simulation and real test are almost the same. With an exception for the parts where the vehicle should drive a constant velocity. There it is visible that the real vehicle has more fluctuation in the velocity. This is a logic result because the simulation has no real vehicle dynamics that are considered. It can also be observed that there is some more over- and undershoot in the test response. Again this is due to the fact that there is no vehicle dynamics involved in the simulation, and therefore the real response is less accurate. Another reason could be that the Adaptive Cruise Control (ACC) that is used for the velocity control isn't perfectly accurate.

Chapter 8

Conclusion

In this thesis a path following system has been developed for the autonomous vehicle of the Delft University of Technology (TU Delft). The design of this path follower is done in different steps. First the necessary vehicle dynamics are reviewed. It is chosen to use a relative simple vehicle model for the lateral design of the system. This bicycle model has two degrees of freedom and although it is a simple model, it is commonly used in literature for path following purposes. For the longitudinal part of the path following system there is no vehicle model used. This is a result of the autonomous vehicle that is used. The inputs of this vehicle are set in the Move-Box, and for the velocity the Move-Box desires a body acceleration value. As explained in chapter 4 this means that there is no use for a vehicle model because the control of a vehicle model starts at the wheel acceleration. Due to the fact that the inside of the Move-Box is unknown it was impossible to design a model for this part of the system.

The vehicle models require parameters, and this was the next step in the development of the path follower. The process of finding these parameters is described in chapter 3. First the parameters that could be measured were acquired. After this process was complete the resulting parameters were optimized with use of a non-linear least squares method. The optimization was done on different data sets that were logged from the Toyota Prius. The best fitted data was selected with use of the variance accounted for (vaf) value. This fitted parameters were than used in the rest of the thesis.

The controller design was discussed in chapter 4. The control algorithms were selected with the aim of this thesis in mind. Therefore the algorithms needed to perform well in path following but also in real-time possibilities. This resulted in a lateral future predictive controller that was also improved in the same chapter. Later in the thesis it was proven that these improvements did enhance the performance of the path follower. For the longitudinal controller it was chosen to use a PD controller that was introduced in literature.

The design of the rest of the system was described in chapter 5. The system was designed with the Autonomous Vehicle in mind. This resulted in a system that was capable to rewrite the outputs of the vehicle to the required inputs of the controller. The path follower was designed

in LabVIEW, which made it possible to quickly implement the system into the Autonomous vehicle. And even though there was limited testing time it was possible to perform quite a number of tests. This was a result of the fast and easy implementation. The system is designed to fit a curve over a part of the trajectory. This curve fitting is done twice, to create redundancy in the system and prevent failing in some extreme conditions. A benefit of this fitting is the possibility of connecting it to a path planner. While driving the system would search the nearest path point from which it would use a small proportion of the actual path. When connected to a path planner the system would be able to work exactly the same, the only difference is that it will no longer be necessary to select a portion of the trajectory. In this thesis there is no path planner used because it is still in development.

After the design of the path following system it was evaluated with use of simulation and real tests. These evaluations are described in chapter 6 and 7. The performance of the controller was analysed and the tuning process was discussed. The conclusion from these chapters is that the performance of the path follower is good. The results of the real vehicle test were almost within the standard deviation of the GPS. The performance of the simulations were worse than of the real tests and this can be a result of vehicle dynamics that were not considered in the bicycle model.

The simulations that were done didn't reach a higher velocity than 50km/h . It was concluded that the path follower would have a larger error for a higher velocity. It was concluded also that this problem could be solved partially by selecting new control values for different velocities. But it is assumed that this wouldn't make this problem disappear completely. Therefore there is an improvement of the current controller introduced in appendix A. The idea of this controller is to generate more controller inputs for higher velocities, which should result in a better performance.

The main aim of this MSc thesis was defined as follows:

Designing and implementing a path following system for an autonomous vehicle that will follow a predefined trajectory in real-time.

In the thesis a path following system was designed, the system was tested and validated in a real autonomous vehicle. The system was able to run real-time, although in testing it was at a lower frequency than intended. This wasn't a problem of the path follow system but some unexplained fault during the tests. This doesn't mean that the system was unable to run on the maximum frequency of the vehicle, 25 Hertz. During testing the CPU Load of the PCI eXtension for Instrumentation (PXI) was checked and there was more than enough processing power left to run at 25 hertz. Unfortunately it was not possible to retest the vehicle due to time limits and other researchers that needed the vehicle.

Nevertheless it can be concluded that the aim of this MSc thesis is met. The results of the real test are that the controller stays within 15.8cm of the path, where as in simulation the error is 1.19cm . The next chapter will discuss some of the next steps that are necessary before the autonomous vehicle will actually drive on the open road.

Further Work Proposals

Before the path follow system can be used for actual driving on the open road there is quite some more research necessary. First one must find the difference between the model and the real vehicle. As discussed before there is a difference between both of the responses, and this difference could be a result of a vehicle model that doesn't evaluate all the vehicle dynamics. Therefore the first proposal is to use a more advanced vehicle model. This can only be done when more parameters of the vehicle are acquired.

When the vehicle model and real vehicle do no longer have differences in their response it is possible to test and simulate for higher velocities, this is necessary to prove the working of the system in all circumstances. If the lateral controller has indeed a larger distance error at higher velocities than the proposed controller must be tested. This proposed controller is explained in appendix A.

The first tests with the path following system have been conducted, and one of the results is that there is quite an amount of computational power of the CPU left. This means that it is now possible to design and use more computationally heavy controller algorithms that can possibly improve the performance of the controller. Especially at higher velocities. This will result in a fuzzy control logic that is probably required for an autonomous vehicle that is capable of driving at all different velocities.

From the figures of chapter 7 it can be concluded that there is not much difference in the response when simulated at different frequencies. As was discussed these results may vary for higher velocities and therefore this should also be researched. Next to that it is also necessary to investigate the influence of this lower frequency on emergency manoeuvres. Although it isn't part of the research of this thesis it can be said that when the vehicle has to perform an emergency manoeuvre every millisecond counts. If the system is capable of working at a frequency of 100 hertz than it can react 4 times faster than when it is working at 25 hertz. This has a large influence on the longitudinal and lateral reaction time and therefore this difference should be investigated when conducting research on evasive manoeuvres.

Before it will be possible to do more research on longitudinal control and these previously described evasive manoeuvres, it will be required to have a better understanding of the Move-Box controller. Then a suitable longitudinal model can be constructed that will be able to simulate the real vehicle response. This model can then be used to design a longitudinal controller that should be able to perform emergency braking.

In this thesis it was concluded that the heading of the path has large influence on the steering response of the path follower. Further research can be done in this heading parameter and the influence of the heading of the path to the actual vehicle response.

The last proposal relates to the position determination. In this thesis the Global Positioning System (GPS) is used to determine the location and heading of the vehicle. This was done at a location with good satellite reception. In other situations this GPS reception can be worse, for instance when driving below trees or between high buildings in cities. Therefore it would be necessary to investigate other options of localisation. It is required to localise the vehicle in every situation before it is possible to have a working autonomous vehicle.

Appendix A

Proposed controller

This appendix will introduce a variant of the lateral controller that was used in this thesis. The design of this controller was inspired by the controller design of [20] and [21]. The idea of this controller and the controllers from literature are quite similar. But the control algorithm are completely different.

The problem with the previously lateral controller, which is described in chapter 4, is that at higher velocity there is a larger distance error y_e to the path. This is, for a part, a result of the feed-forward length L_f . This length becomes larger when the velocity increases. This larger feed-forward is required for the controller to work and to be stable, but it does also result in a control distance y_{ef} that is further and further away from the vehicle. This larger feed-forward will eventually lead to cutting corners, even though the control parameters are chosen perfectly.

To solve this problem a change in the control algorithm is proposed. Instead of calculating one distance y_{ef} a couple of distances will be calculated, as shown in Figure A-1. In that way the controller would have more information of the path, and therefore it should have a better tracking.

The distance between the lines will be defined as the forward velocity and a control parameter T . This T depend on the feed-forward term k_f and the number of error lines y_{efN} :

$$T = \frac{k_f}{y_{efN}} \quad (\text{A-1})$$

With this definition the distance between the lines will scale with the forward velocity, as is the same principle as with the lateral controller that was introduced before. But of course it is possible to scale the number of error calculations. The maximum number of lines depend on the calculation power of the PCI eXtension for Instrumentation (PXI), or the maximum is reached when the controller response doesn't improve any more. The one that comes first results in the maximum number.

In the previous controller there was one control parameter for the distance, the control parameter k_s . To maintain the same tuning principle it is chosen to define the same control

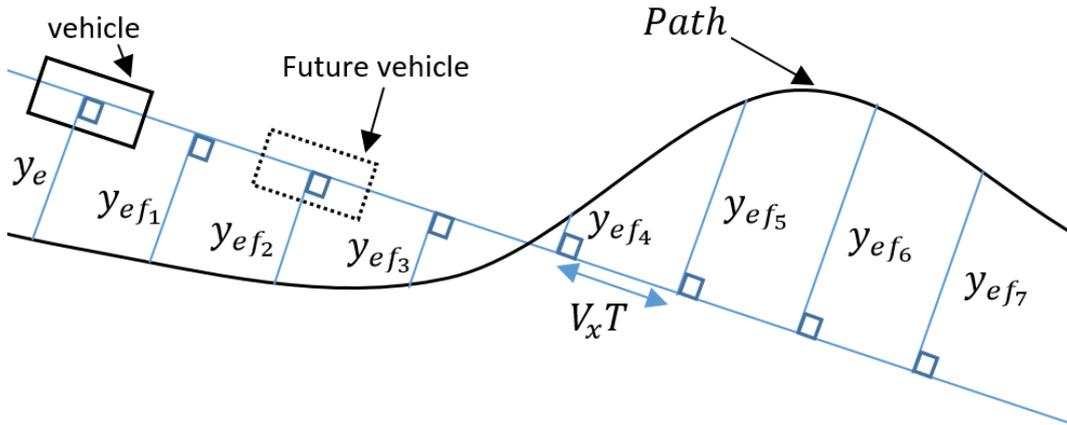


Figure A-1: The preview structure of the proposed controller algorithm

parameter as weight function for this second controller. But it is not possible to multiply this control parameter with every value y_{efn} , this would result in a very unstable system. Therefore the error distances need be given a weight.

Three different control modus are introduced in Figure A-2. The first control mode will give the same weight factor to all the distances. The second mode has an increasing weight for points that are further away from the vehicle. The last mode has the highest weight value for the closest error distance. The Control law for this control method is given in equation A-2. For stability reasons the weight factor will be summed to the value of k_s .

$$\delta = \sin(\theta_e) + \frac{\sum D_e}{v_x} \quad (\text{A-2})$$

| | | | |
|----------------------------------|--|--------------------------|---|
| Mode 1 Equally divided | | $k_{sn} = \frac{k_s}{N}$ | $D_e(n) = k_{sn} * y_{ef}(n)$ |
| Mode 2 Increasing | | $k_{sn} = \frac{k_s}{N}$ | $D_e(n) = k_{sn} * y_{ef}(n) * n$ |
| Mode 3 Decreasing | | $k_{sn} = \frac{k_s}{N}$ | $D_e(n) = k_{sn} * (N - n + 1) * y_{ef}(n)$ |

Figure A-2: The different control modus and their calculations

Bibliography

- [1] A. Idriz, *Safe Interaction Between Lateral and Longitudinal Adaptive Cruise Control in Autonomous Vehicles*. PhD thesis, TU Delft, Delft University of Technology, 2015.
- [2] T. D. Gillespie, *Fundamentals of vehicle dynamics*. Society of Automotive Engineers (SAE), 1992.
- [3] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.
- [4] M. Awan, *Compensation of low performance steering system using torque vectoring*. PhD thesis, Cranfield University, 2012.
- [5] R. Isermann, “Diagnosis methods for electronic controlled vehicles,” *Vehicle System Dynamics*, vol. 36, no. 2-3, pp. 77–117, 2001.
- [6] J. Gómez Fernández, “A vehicle dynamics model for driving simulators,” 2012.
- [7] H. B. Pacejka, E. Bakker, and L. Nyborg, “Tyre modelling for use in vehicle dynamics studies,” *SAE paper*, vol. 870421, pp. 1–15, 1987.
- [8] H. Dugoff, P. Fancher, and L. Segel, “An analysis of tire traction properties and their influence on vehicle dynamic performance,” tech. rep., SAE Technical Paper, 1970.
- [9] Ivo Houtzager, “LTI System Identification Toolbox.” http://www.dcsc.tudelft.nl/~jwvanwingerden/lti/ltitoolbox_product_page.html. Accessed: 21-11-16.
- [10] S. Arrigoni, F. Cheli, S. Manazza, P. Gottardis, R. Happee, M. Arat, and D. Kotiadis, “Autonomous vehicle controlled by safety path planner with collision risk estimation coupled with a non-linear mpc,” in *The Dynamics of Vehicles on Roads and Tracks: Proceedings of the 24th Symposium of the International Association for Vehicle System Dynamics (IAVSD 2015), Graz, Austria, 17-21 August 2015*, p. 199, CRC Press, 2016.
- [11] A. De Luca, G. Oriolo, and C. Samson, “Feedback control of a nonholonomic car-like robot,” in *Robot motion planning and control*, pp. 171–253, Springer, 1998.

- [12] 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.
- [13] J. M. Snider, “Automatic steering methods for autonomous automobile path tracking,” *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RITR-09-08*, 2009.
- [14] J. Xu, K. Yang, Y. Shao, and G. Lu, “An experimental study on lateral acceleration of cars in different environments in sichuan, southwest china,” *Discrete Dynamics in Nature and Society*, vol. 2015, 2015.
- [15] M. A. Zakaria, H. Zamzuri, R. Mamat, and S. A. Mazlan, “A path tracking algorithm using future prediction control with spike detection for an autonomous vehicle robot,” *International Journal of Advanced Robotic Systems*, vol. 10, 2013.
- [16] R. C. Coulter, “Implementation of the pure pursuit path tracking algorithm,” tech. rep., DTIC Document, 1992.
- [17] S. Moon and K. Yi, “Human driving data-based design of a vehicle adaptive cruise control algorithm,” *Vehicle System Dynamics*, vol. 46, no. 8, pp. 661–690, 2008.
- [18] S. Moon, I. Moon, and K. Yi, “Design, tuning, and evaluation of a full-range adaptive cruise control system with collision avoidance,” *Control Engineering Practice*, vol. 17, no. 4, pp. 442–455, 2009.
- [19] Robert, “GPS position to UTM coordinates VI.” <https://forums.ni.com/t5/Community-Documents/GPS-position-to-UTM-coordinates/ta-p/3536992>. Accessed: 08-11-16.
- [20] R. S. Sharp, “Driver steering control and a new perspective on car handling qualities,” *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 219, no. 10, pp. 1041–1051, 2005.
- [21] R. Sharp and V. Valtetsiotis, “Optimal preview car steering control,” *Vehicle System Dynamics*, vol. 35, pp. 101–117, 2001.

Glossary

List of Acronyms

| | |
|-----------------|---|
| TU Delft | Delft University of Technology |
| ACC | Adaptive Cruise Control |
| ADAS | Advanced Driving Assist Systems |
| CoG | Centre of Gravity |
| DAVI | Dutch Automated Vehicle Initiative |
| GPS | Global Positioning System |
| PXI | PCI eXtension for Instrumentation |
| RDW | Rijksdienst voor het Wegverkeer |
| RMS | Root Mean Square |
| TTC | Time To Collision |
| UTM | Universal Transverse Mercator coordinate system |
| vaf | variance accounted for |
| VI | Virtual Instrument |

List of Symbols

| | |
|----------------|--|
| α_f | Slip angle of the front tyre |
| α_r | Slip angle of the rear tyre |
| γ | Path Angle |
| δ | Steering angle |
| θ_e | Heading error |
| θ_p | Desired heading angle form the path |
| λ | Longitudinal tyre slip |
| τ_δ | Dynamic steering time constant |
| ω | Wheel rotations |
| a_c | Acceleration value |
| a_y | Lateral acceleration in local frame |
| $C_{\alpha f}$ | Cornering stiffness of 2 front tyres |
| $C_{\alpha r}$ | Cornering stiffness of 2 rear tyres |
| (c_x, c_y) | Current vehicle position |
| f_x, f_y | Future point |
| k_δ | Steering ratio |
| K_d | Derivative gain |
| K_f | Control parameter for look-ahead distance |
| k_h | Control parameter for heading error |
| K_p | Proportional gain |
| k_{path} | Control parameter for length of path section |
| k_s | Control parameter for steering |
| K_{us} | Under-steer gradient |
| L | Wheelbase of the vehicle |
| L_f | Look-ahead distance |
| l_f | Length between CoG and the front axle |
| l_r | Length between CoG and the rear axle |
| L_p | Length of the path measured from starting point |
| m | Vehicle mass |
| P_n | Path point |
| (p_x, p_y) | Point on the path perpendicular from the vehicle |
| \dot{r} | Angular acceleration |
| r | Yaw rate or angular velocity |
| R_e | Wheel radius |
| T_0 | First heading point of local path |
| v_c | Current vehicle velocity |
| v_e | Velocity error |
| v_p | Desired velocity from the path |

| | |
|-------------------|---------------------------------------|
| v_r | Wheel velocity |
| v_{sx} | Longitudinal slip velocity |
| v_x | Longitudinal velocity |
| \dot{v}_y | Lateral acceleration |
| v_y | Lateral velocity |
| \dot{X} | Longitudinal velocity in global frame |
| (X, Y, Z) | Global vehicle positions |
| (X_l, Y_l, Z_l) | Local vehicle positions |
| X_n | Normalised X values |
| X_0 | First X point of local path |
| \dot{Y} | Lateral velocity in global frame |
| Y_0 | First Y point of local path |
| y_e | Lateral perpendicular error distance |
| y_{ef} | Furture lateral error |
| Y_n | Normalised Y values |

